

# **UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO**

## **FACULTAD DE INGENIERÍAS SEDE QUITO - CAMPUS SUR**

### **CARRERA DE INGENIERÍA DE SISTEMAS MENCIÓN TELEMÁTICA**

#### **TÍTULO O TEMA DEL TRABAJO**

Diseño, implementación y monitoreo de una red Mesh/Zigbee aplicado a la domótica, gestionada mediante SMS (Short Message Service), servidor web, correo electrónico, aplicación bluetooth para dispositivos móviles y control inalámbrico local.

#### **TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS**

#### **AUTORES**

**MIGUEL ÁNGEL CEDEÑO VILLARROEL, CARLOS EDUARDO ERAS PÉREZ**

#### **DIRECTOR**

**RAFAEL JAYA**

**QUITO, noviembre 2010**

## **DECLARACIÓN**

Nosotros, Miguel Ángel Cedeño Villarroel, Carlos Eduardo Eras Pérez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

---

Miguel Ángel Cedeño Villarroel

---

Carlos Eduardo Eras Pérez

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Miguel Ángel Cedeño Villarroel, Carlos Eduardo Eras Pérez bajo mi dirección.

---

Rafael Jaya

Director de Tesis

## **DEDICATORIA**

La presente tesis se la dedico a mis padres y a toda mi familia, gracias a su apoyo constante e incondicional logré terminar un objetivo primordial en mi vida ser Ingeniero, la constante dedicación de mis padres fue pilar fundamental para encarrilar mi vida hacia un futuro mejor a través de sus consejos, enseñanzas y amor.

A mi abuela que aun nos acompaña y a mis abuelos que ya no están aquí, a mis tías por la ayuda en momentos de cambio brindándome estadía y alimentación siempre cuando la necesité.

A los profesores que gracias a su paciencia lograron hacer de mi un buen profesional, al tutor de tesis que apoyó de sobremanera este tema e hizo posible su realización, a mis verdaderos amigos/as que siempre me felicitaban y aconsejaban por cada pequeño logro alcanzado en mi vida.

Miguel Ángel Cedeño Villarroel



## **AGRADECIMIENTO**

A Dios por darme la oportunidad de vivir y tener una familia maravillosa.

Con mucho cariño principalmente a mis padres que me dieron la vida y la oportunidad de estudiar, gracias de corazón por creer en mí y darme una carrera para mi futuro, aunque hemos pasado momentos difíciles siempre han estado ahí apoyándome y brindándome todo su amor incondicional.

Este trabajo que me llevo casi un año realizarlo es para ustedes.

A mis profesores especialmente a mi tutor, con su ayuda y dedicación se pudo llevar a cabo esta idea, en el Ecuador si es posible desarrollar tecnología y de muy buena calidad.

A mis amigos/as por darme ánimos en momentos difíciles, y a todas las personas que directa o indirectamente apoyaron con la realización de este proyecto.

Sin ustedes a mi lado no lo hubiera logrado, tantas desveladas sirvieron de algo y aquí está el fruto de ese arduo trabajo, Muchas gracias por todo.

Miguel Ángel Cedeño Villarroel

## **DEDICATORIA**

Al culminar la etapa universitaria y dejar plasmado en estas hojas todos los conocimientos adquiridos durante la carrera, sabiendo que termina una etapa y comienza otra, en la cual los retos se vuelven cada vez más duros, y difíciles de afrontar en un mundo cada vez más competitivo.

La presente tesis tiene una dedicación especial a mi familia, y a mi hijo el cual fue la inspiración y la fuerza que me alentaban a continuar y no decaer en el camino. Llegue a comprender que una persona tan pequeña puede ser el motor que mueva mi vida y me aliente a concluir todas las metas que me proponga, es así que el empeño, sacrificio, puesto en este trabajo tendrá una recompensa, la cual la veré todos los días de mi vida reflejada en la sonrisa de mi hijo y en el orgullo de mis queridos padres.

Carlos Eduardo Eras Pérez

## **AGRADECIMIENTO**

Son varias personas a las cuales debo este triunfo, el lograr culminar una etapa académica más en mi vida, el saber que ellas fueron parte fundamental a lo largo de éstos años, me hace comprender que la meta alcanzada no es solo mía sino de todos ellos también.

En primer lugar a mis padres, y a mi hermana que con su apoyo, dedicación, paciencia, consejos, enseñanza supieron hacer de mí una persona responsable, dedicada y honesta, que pese a todas las adversidades que se pudieron presentar, estuvieron siempre conmigo a todo momento, en especial cuando necesitaba de ellos.

A la persona que estuvo conmigo durante y a lo largo de éste duro camino, gracias a ella que supo comprender y valorar mi esfuerzo, le agradezco no solo los momentos agradables sino los momentos difíciles los cuales nos hicieron más fuertes y maduros, gracias Anita.

De igual manera mi agradecimiento más sincero a todos y cada uno de los profesores, que durante mi carrera universitaria supieron infundir en mi los conocimientos necesarios, que serán las piezas fundamentales para mi desenvolvimiento profesional.

Sin ánimo de olvidar a nadie en particular y a todas aquellas personas que de una u otra manera han compartido mi vida durante el transcurso de éste tiempo mis más sinceros agradecimientos por su amistad, estímulo y ayuda ya que sin ellos éste logro no hubiera sido posible.

Carlos Eduardo Eras Pérez

## **ABSTRACT**

La creciente inseguridad y la necesidad de mayor confort, obligan a buscar nuevos e innovadores métodos de protección, la constante preocupación de no tener vigilada y controlada una residencia en todo momento y lugar, sin saber con certeza que es lo que ocurre en su interior inspiró el desarrollo de un sistema domótico capaz de satisfacer estas necesidades.

El uso de dispositivos móviles es muy común en nuestra vida cotidiana, por esta razón el integrarlos a la seguridad del hogar garantiza una mayor disponibilidad de monitoreo y gestión de tareas rutinarias como encender/apagar luces, abrir/cerrar cortinas, verificar el estado en tiempo real de sensores tales como detectores de movimiento, detectores de gas, detectores de humo, sensor fotovoltaico, etc.

El sistema domótico que se plantea en esta tesis permite el control y monitoreo del hogar por medio de: SMS (servicio de mensajes cortos) enviando un mensaje al servidor SMS con la petición que se necesite, Aplicación .JAR Bluetooth en un dispositivo celular móvil conectado con el servidor, Control inalámbrico local Touch desarrollado en un GLCD 240x128 pixeles con un panel resistivo táctil, Implementación de aplicación WEB desarrollada en java y alojada en un servidor Tomcat publicado por medio de un hosting gratuito como DynDNS y Cámara IP mostrada en la aplicación web java.

Los dispositivos sensores y actuadores están divididos en 3 módulos cada uno gestionado por un microcontrolador y usando un módulo Xbee serie 2 para su comunicación con el servidor principal.

La comunicación entre los módulos Xbee se realiza usando la topología Mesh, permitiendo una rápida convergencia de los clientes Routers para establecer comunicación con el coordinador.

# ÍNDICE

## CAPÍTULO 1

1.	ANTECEDENTES	25
1.1.	PROBLEMA	25
1.2.	OBJETIVOS	25
1.2.1.	GENERAL	25
1.2.2.	ESPECÍFICOS	26
1.3.	JUSTIFICACIÓN	26
1.4.	ALCANCE	27
1.4.1.	SEGURIDAD FÍSICA	28
1.4.2.	SEGURIDAD LÓGICA	28
1.4.2.1.	En la aplicación Web	28
1.4.2.2.	En la aplicación bluetooth	28
1.4.2.3.	SMS (Short Message Service)	29
1.4.2.4.	Confort	29
1.4.2.5.	Simulación de presencia	29
1.4.2.6.	Modo ahorro de energía	29
1.4.2.7.	Bases de Datos	30
1.4.3.	HISTÓRICOS	30

## CAPÍTULO 2

2.	SUSTENTO TEÓRICO	31
2.1.	DOMÓTICA	31
2.1.1.	PILARES FUNDAMENTALES	31
2.1.1.1.	Seguridad de bienes y personas	31
2.1.1.2.	Gestión y ahorro de energía	32
2.1.1.3.	Confort y entretenimiento dentro del hogar	32
2.1.1.4.	Comunicación y trabajo	32
2.1.2.	ARQUITECTURAS DE UN SISTEMA DOMÓTICO	32
2.1.2.1.	Arquitectura centralizada	32
2.1.2.2.	Arquitectura distribuida	33
2.1.2.3.	Arquitectura mixta	33
2.2.	TECNOLOGÍA ZIGBEE	33
2.2.1.	ARQUITECTURA DE UNA RED ZIGBEE	34
2.2.1.1.	Capas de una Red Zigbee	35
2.2.1.1.1.	Capa Física	35
2.2.1.1.2.	Capa de MAC	35

2.2.1.1.3.	Capa de red NWK (Network Layer)	35
2.2.1.1.4.	Capa Aplicación (APL Application Layer)	35
2.2.2.	ELEMENTOS DE UNA RED ZIGBEE	36
2.2.2.1.	Dispositivo Coordinador	36
2.2.2.2.	Router	37
2.2.2.3.	End Device	37
2.2.3.	TOPOLOGÍAS DE RED ZIGBEE	37
2.2.3.1.	Star o Estrella	37
2.2.3.2.	Mesh o Malla	38
2.2.3.3.	Clúster Tree o Racimo de Árbol	38
2.2.4.	ESTRATEGIAS DE CONEXIÓN EN UNA RED ZIGBEE	38
2.2.4.1.	Con balizas	39
2.2.4.2.	Sin balizas	39
2.3.	MÓDULOS DE COMUNICACIONES	41
2.3.1.	MÓDULOS GSM	41
2.3.1.1.	GSM - SIM Card	41
2.3.1.2.	Seguridad GSM	42
2.3.1.2.1.	Código PIN	42
2.3.1.2.2.	Código PUK	42
2.3.1.3.	Comandos AT con GSM	42
2.3.2.	MÓDULOS ZIGBEE	43
2.3.2.1.	Tipos de Módulos Xbee	45
2.3.2.1.1.	Módulos Xbee	45
2.3.2.1.2.	Módulos Xbee Pro	45
2.3.2.2.	Requerimientos de Conexión	46
2.3.2.3.	Seguridad en los Módulos Xbee	46
2.4.	DISPOSITIVOS ELECTRÓNICOS	47
2.4.1.	MICROCONTROLADORES	47
2.4.1.1.	Componentes del microcontrolador	48
2.4.1.1.1.	El UCP o procesador	48
2.4.1.1.2.	Memoria	49
2.4.1.1.3.	Puertas o Líneas de Entrada y Salida	51
2.4.1.1.4.	Reloj Principal	51
2.4.1.1.5.	Recursos Adicionales	51
2.4.1.2.	Arquitectura de Microcontroladores	53
2.4.1.2.1.	Arquitectura Von Neumann	53
2.4.1.2.2.	Arquitectura Harvard	53
2.4.2.	SENSORES Y ACTUADORES	54
2.4.2.1.	Sensor magnético	54
2.4.2.1.1.	Partes de un sensor magnético:	54

2.4.2.1.2. Principio de Funcionamiento	55
2.4.2.2. Sensor Fotovoltaico	55
2.4.2.2.1. Fotorresistencias	55
2.4.2.2.2. Células fotovoltaicas	56
2.4.2.3. Detector de movimiento	56
2.4.2.3.1. Microondas	56
2.4.2.3.2. Detectores Infrarrojos Pasivos de Movimiento	57
2.4.2.4. Detector de humo	58
2.4.2.4.1. Sensor de humo por efecto fotoeléctrico	58
2.4.2.4.2. Sensor de humo por ionización	59
2.4.2.5. Detector de gas	59
2.4.2.6. Relés	60
2.4.3. PROTOTIPO CORTINAS ELÉCTRICAS	60
2.4.4. CÁMARA IP	61
2.4.4.1. Componentes de la Cámara IP	61
2.4.5. PANTALLA TÁCTIL	62
2.4.5.1. Funcionamiento de una Pantalla Táctil	62
2.5. SOFTWARE DE DESARROLLO	63
2.5.1. LENGUAJE DE PROGRAMACIÓN	63
2.5.1.1. Tipos de Lenguajes de Programación	64
2.5.1.1.1. Lenguajes interpretados	64
2.5.1.1.2. Lenguajes compilados	64
2.5.1.1.3. Lenguajes intermediarios	64
2.5.2. BASE DE DATOS	65
2.5.3. SERVIDORES	66
2.5.3.1. Servidor Web	66
2.5.3.2. Servidor de correo Web (Webmail)	67
CAPÍTULO 3	68
3. ANÁLISIS DE REQUERIMIENTOS	68
3.1. ANÁLISIS DE TECNOLOGÍAS INALÁMBRICAS	68
3.1.1. BLUETOOTH	69
3.1.2. INFRARROJO	69
3.1.2.1. Tipos de comunicación infrarroja	70
3.1.2.1.1. Cuasi difuso	70
3.1.2.1.2. Difuso	70
3.1.3. ZIGBEE	71
3.2. ANÁLISIS DE SOFTWARE	72
3.2.1. GESTOR DE BASES DE DATOS	72
3.2.1.1. SQL Server 2008	73

3.2.1.2.	Oracle	74
3.2.1.3.	MySQL	75
3.2.1.4.	PostgreSQL	76
3.2.2.	LENGUAJES DE PROGRAMACIÓN APLICACIÓN WEB	78
3.2.2.1.	Lenguaje PHP	78
3.2.2.2.	Lenguaje Java	79
3.2.2.3.	Lenguaje Visual Basic.Net 2008	82
3.2.3.	SERVIDORES WEB	84
3.2.3.1.	IIS	84
3.2.3.2.	Apache	85
3.2.3.3.	Cherokee	86
3.2.4.	SERVIDORES DE CORREO WEB	87
3.2.4.1.	Hotmail	87
3.2.4.2.	GMAIL	88
3.2.5.	LENGUAJES DE PROGRAMACIÓN DE MICROCONTROLADORES	90
3.2.5.1.	Lenguaje BASIC	90
3.2.5.2.	Lenguaje C	90
3.2.5.3.	Lenguaje Ensamblador (assembler)	91
3.2.6.	HERRAMIENTAS DE DESARROLLO JAVA	92
3.2.6.1.	Eclipse	92
3.2.6.2.	Netbeans	92
3.2.6.3.	JCreator Pro	92
3.2.6.4.	BlueJ	92
3.2.6.5.	AutoJava	92
3.2.7.	HERRAMIENTAS DE DESARROLLO BASIC PARA MICROCONTROLADORES	93
3.2.7.1.	MikroBasic	93
3.2.7.2.	Microcode Studio	93
3.2.8.	HERRAMIENTAS DE DESARROLLO APLICACIONES MÓVILES BLUETOOTH	94
3.2.8.1.	Mobile Processing	94
3.2.8.2.	J2ME Wireless Toolkit	95
3.2.9.	SIMULADORES DE CIRCUITOS ELECTRÓNICOS	95
3.2.9.1.	Proteus	96
3.3.	ANÁLISIS DE HARDWARE	97
3.3.1.	MÓDULO GSM	97
3.3.1.1.	Servicios GSM	97
3.3.1.1.1.	SMS	98
3.3.1.1.2.	Mensajes multimedia	98
3.3.2.	MÓDULOS ZIGBEE	99
3.3.3.	ANÁLISIS GLCD	102
3.3.3.1.	Tipos de GLCD	102



3.3.3.1.1.	Infrarrojos	102
3.3.3.1.2.	Resistivas	103
3.3.3.1.3.	Pantallas táctiles capacitivas	103
3.3.4.	ANÁLISIS DE SENSORES	104
3.3.4.1.	Sensor Magnético	104
3.3.4.2.	Sensor Fotovoltaico	105
3.3.4.3.	Sensores de Movimiento	106
3.3.4.4.	Detector de Humo	107
3.3.4.5.	Detector de Gas	108
3.3.5.	ANÁLISIS DE ACTUADORES	109
3.3.5.1.	Sirena	109
3.3.5.2.	Cámara IP	110
3.3.5.3.	Modular de simulación de apertura y cierra de cortinas	112
3.3.5.4.	Inversor de Giro de un Motor	113
3.3.5.4.1.	Puente H con Transistores	113
3.3.5.4.2.	Inverso de Giro con Relés	113
3.3.6.	ANÁLISIS DE MICROCONTROLADORES	114
3.3.7.	SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA SAI - UPS	116
3.4.	CONEXIÓN A INTERNET	118
3.5.	REQUERIMIENTOS BÁSICOS DEL SERVIDOR	119
3.5.1.	REQUERIMIENTOS MÍNIMOS	119
3.5.2.	REQUERIMIENTOS RECOMENDADOS	119
CAPITULO 4		
120		
4.	DISEÑO DEL PROYECTO	
120		
4.1.	CRITERIOS PARA EL DISEÑO DE LA SOLUCIÓN	120
4.1.1.	CRITERIOS TÉCNICOS	120
4.1.2.	CRITERIOS DEL USUARIO	120
4.2.	DISEÑO DE LA SOLUCIÓN	122
4.2.1.	ZIGBEE	124
4.2.1.1.	Módulos de comunicación	124
4.2.1.2.	Nomenclatura de identificación	125
4.2.1.3.	Códigos de comunicación	129
4.2.1.3.1.	Enviados hacia el servidor	129
4.2.1.3.2.	Enviados desde el servidor	133
4.2.2.	BLUETOOTH	134
4.2.2.1.	Interface de comunicación	134
4.2.2.2.	Códigos de comunicación	136

4.2.2.2.1. Enviados hacia el servidor	136
4.2.2.2.2. Enviados desde el servidor	137
4.2.3. SMS	138
4.2.3.1. Comandos de comunicación	139
4.2.3.1.1. Enviados hacia el servidor	139
4.2.3.1.2. Enviados desde el servidor	140
4.2.4. Correo Electrónico	142
4.2.5. Cámara IP	143
4.2.6. Servidor Principal	143
4.3. DISEÑO DE HARDWARE	146
4.3.1. MÓDULO XBEE1	146
4.3.1.1. Actuadores	148
4.3.1.2. Sensores	149
4.3.2. MÓDULO XBEE2	149
4.3.2.1. Actuadores	151
4.3.2.2. Sensores	151
4.3.3. MÓDULO XBEE3	152
4.3.3.1. Actuadores	154
4.3.3.2. Sensores	154
4.3.4. MÓDULO GLCD	155
4.3.5. MÓDULO COORDINADOR	157
4.4. DISEÑO DE BASE DE DATOS	158
4.4.1. MODELO FÍSICO DE LA BASE DE DATOS	159
4.4.2. MODELO CONCEPTUAL DE LA BASE DE DATOS	160
4.4.3. DIAGRAMA BASE DE DATOS	161
4.4.4. DICCIONARIO DE DATOS	162
4.5. DISEÑO DE SOFTWARE	166
4.5.1. MICROCONTROLADORES	166
4.5.1.1. Módulo Xbee1	166
4.5.1.1.1. Diagrama de Flujo	166
4.5.1.2. Módulo Xbee2	168
4.5.1.2.1. Diagrama de Flujo	169
4.5.1.3. Módulo Xbee3	170
4.5.1.3.1. Diagrama de Flujo	171
4.5.1.4. Módulo GLCD Touch	172
4.5.1.4.1. Diagrama de Flujo	177
4.5.1.4.2. Casos de uso	178
4.5.1.4.3. Diagrama de clases (módulos)	182
4.5.2. APLICACIÓN MÓVIL BLUETOOTH J2ME	183
4.5.2.1. Diagrama de bloques	187

4.5.2.2.	Diagrama de Flujo	188
4.5.2.3.	Casos de uso	189
4.5.2.4.	Diagrama de clases	193
4.5.3.	SERVIDOR PRINCIPAL	194
4.5.3.1.	Módulos servidor principal	195
4.5.3.1.1.	Módulo Servidor SMS	195
4.5.3.1.2.	Módulo Servidor Bluetooth	197
4.5.3.1.3.	Módulo Servidor Zigbee	198
4.5.3.1.4.	Módulo Servidor Correo	199
4.5.3.1.5.	Módulo Servidor Cámara IP	201
4.5.3.1.6.	Módulo Servidor Base de datos	201
4.5.3.2.	Diagrama de navegación	203
4.5.3.3.	Estructura de Páginas	203
4.5.3.4.	Bocetos de diseño	204
4.5.3.5.	Mapa permanente del sitio	205
4.5.3.6.	Diagrama de Flujo	206
4.5.3.7.	Casos de uso	206
4.5.3.8.	Diagrama de clases	206
4.5.3.9.	Seguridades	206
4.5.3.9.1.	Control de Usuarios	207
4.5.3.9.2.	RespalDOS Base de datos	207
4.5.3.9.3.	Encriptación Base de datos	207
CAPITULO 5		208
5.	IMPLEMENTACIÓN	208
5.1.	CONFIGURACIONES	208
5.1.1.	ZIGBEE	208
5.1.2.	BLUETOOTH	217
5.2.	DESARROLLO DE HARDWARE	221
5.2.1.	CIRCUITOS BÁSICOS	221
5.2.1.1.	Conexión módulo Zigbee – Microcontrolador	222
5.2.1.2.	Multiplexor de voltaje	223
5.2.1.3.	Detector de energía	224
5.2.1.4.	Detector de Luz	225
5.2.1.5.	Detector de Gas GLP	228
5.2.1.6.	Inversor de giro Motor DC	232
5.2.2.	MÓDULO XBEE1	236
5.2.3.	MÓDULO XBEE2	244
5.2.4.	MÓDULO XBEE3	252
5.2.5.	MÓDULO GLCD	260

5.2.6.	MÓDULO COORDINADOR	268
5.3.	DESARROLLO DE SOFTWARE	272
5.3.1.	MICROCONTROLADORES	272
5.3.1.1.	Módulo Xbee1	272
5.3.1.2.	Módulo Xbee2	278
5.3.1.3.	Módulo Xbee3	281
5.3.1.4.	Módulo GLCD Touch	285
5.3.2.	APLICACIÓN MÓVIL BLUETOOTH J2ME	301
5.3.3.	SERVIDOR PRINCIPAL	310
5.3.3.1.	Estándares de programación	310
5.3.3.1.1.	Estándar de variables	311
5.3.3.1.2.	Estándar de componentes de la interfaz	311
5.3.3.2.	Módulos	312
5.3.3.2.1.	Módulo Servidor SMS	312
5.3.3.2.2.	Módulo Servidor Bluetooth	316
5.3.3.2.3.	Módulo Servidor Zigbee	322
5.3.3.2.4.	Módulo Servidor Correo	335
5.3.3.2.5.	Módulo Servidor Cámara IP	339
5.3.3.3.	Interfaces	341
5.3.3.3.1.	Inicio de sesión	342
5.3.3.3.2.	Monitorear	347
5.3.3.3.3.	Control Iluminación	350
5.3.3.3.4.	Control Cerraduras	351
5.3.3.3.5.	Control Cortina – Bocina	353
5.3.3.3.6.	Control Modos	355
5.3.3.3.7.	Servicios	360
5.3.3.3.8.	Notificaciones SMS	362
5.3.3.3.9.	Notificaciones Correo	364
5.3.3.3.10.	Notificaciones Guardar Historial	367
5.3.3.3.11.	Gestión Usuarios	368
5.3.3.3.12.	Gestión Base de datos	382
5.3.3.4.	Seguridades	397
5.3.3.5.	Encriptación	397
5.3.3.6.	Respaldos	398
CAPITULO 6		399
6.	PRUEBAS	399
6.1.	PRUEBAS DE COMPATIBILIDAD	399
6.2.	PRUEBAS DE SEGURIDAD	401
6.3.	PRUEBAS DE LA BASE DE DATOS	401

6.4.	PRUEBAS DE CONECTIVIDAD ZIGBEE	401
6.5.	PRUEBAS DE FUNCIONAMIENTO	407
6.5.1.	BLUETOOTH	407
6.5.2.	MÓDULO XBEE1	408
6.5.3.	MÓDULO XBEE2	409
6.5.4.	MÓDULO XBEE3	410
6.5.5.	MÓDULO GLCD	411
	CONCLUSIONES	412
	RECOMENDACIONES	414
	BIBLIOGRAFÍA	416

## ÍNDICE DE FIGURAS

Figura 1:	Diagrama de Flujo Transmisión Mediante Xbee	40
Figura 2:	Tipos de Módulos Xbee	45
Figura 3:	Tipos de Módulos XbeePro	45
Figura 4:	Fotografía Módulo Xbee Serie2	101
Figura 5:	Fotografía GLCD	104
Figura 6:	Fotografía Sensor Magnético	105
Figura 7:	Fotografía LDR	105
Figura 8:	Fotografía Detector de Movimiento	107
Figura 9:	Fotografía Detector de Humo	108
Figura 10:	Fotografía Detector de Gas GLP	109
Figura 11:	Fotografía Bocina	110
Figura 12:	Fotografía Cámara IP	112
Figura 13:	Fotografía Prototipo Cortina Eléctrica	113
Figura 14:	Fotografía PIC 18F452	116
Figura 15:	Fotografía Batería 6V – 1.3A	118
Figura 16:	Fotografía Batería 9V	118
Figura 17:	Distribución de los Xbee	122
Figura 18:	Diseño de la solución	123
Figura 19:	Módulo Coordinador	124
Figura 20:	Nomenclatura	129
Figura 21:	Fotografía Bluetooth Dlink USB	135
Figura 22:	Comunicación Bluetooth	135

Figura 23: Comunicación SMS	138
Figura 24: Comunicación con el servidor de correo	142
Figura 25: Diagrama de bloques servidor principal	145
Figura 26: Diagrama de Bloques Módulo Xbee 1	147
Figura 27: Diagrama de Bloques Módulo Xbee 2	150
Figura 28: Diagrama de Bloques Módulo Xbee 3	153
Figura 29: Diagrama de Bloques Módulo GLCD	156
Figura 30: Diagrama de bloques Módulo Coordinador	157
Figura 31: Modelo Físico de la Base de datos	159
Figura 32: Modelo Conceptual de la Base de datos	160
Figura 33: Diagrama de la Base de datos	161
Figura 34: Diagrama de Flujo Módulo Xbee 1	167
Figura 35: Diagrama de Flujo Módulo Xbee 2	169
Figura 36: Diagrama de Flujo Módulo Xbee 3	171
Figura 37: Grafico de navegación administrador	172
Figura 38: Grafico de navegación invitado	173
Figura 39: Pantalla tipo de usuario	173
Figura 40: Pantalla ingreso código de usuario	174
Figura 41: Pestañas para navegación	174
Figura 42: Pestaña control de iluminación	174
Figura 43: Pestaña Monitoreo de sensores	175
Figura 44: Pestaña control de cerraduras	176
Figura 45: Diagrama de Flujo GLCD	177
Figura 46: Casos de uso GLCD	178
Figura 47: Diagrama de Clases GLCD	182
Figura 48: Pantalla inicial Aplicación Bluetooth	184
Figura 49: Pantalla Búsqueda de servidores Puertos disponibles	184
Figura 50: Pantalla Puertos encontrados	185
Figura 51: Pantalla inicial Control de usuario	185
Figura 52: Pantalla Control de iluminación y monitoreo Aplicación Bluetooth	186
Figura 53: Diagrama de Bloques Aplicación Bluetooth	187
Figura 54: Diagrama de Flujo Aplicación Bluetooth	188
Figura 55: Casos de uso Aplicación Bluetooth	189
Figura 56: Diagrama de Clases Aplicación Bluetooth	193
Figura 57: Diagrama de Bloques puertos que usa el servidor	194
Figura 58: Diagrama de Bloques módulo servidor SMS	196
Figura 59: Diagrama de Bloques módulo servidor Bluetooth	197
Figura 60: Diagrama de Bloques módulo servidor Zigbee	198
Figura 61: Diagrama de Bloques módulo servidor Correo	200

Figura 62: Diagrama de Bloques módulo servidor Cámara IP	201
Figura 63: Boceto de diseño Ingreso de usuario	204
Figura 64: Boceto de diseño Portada Principal	204
Figura 65: Árbol funcional usuario tipo ADMINISTRADOR	205
Figura 66: Árbol funcional usuario tipo INVITADO	206
Figura 67: Pantalla principal del XCTU	208
Figura 68: Verifica si el modem Xbee está conectado	209
Figura 69: Muestra pestaña Modem Configuration	209
Figura 70: Obteniendo nuevas versiones de software Xbee	210
Figura 71: Configuración de funciones Xbee	210
Figura 72: Parámetros para configurar el coordinador	211
Figura 73: Configuración de funciones Xbee	212
Figura 74: Parámetros para configurar el Router o end device	213
Figura 75: Prueba de conexión básica entre el Router y el coordinador Xbee	215
Figura 76: Parámetros de configuración de seguridad inalámbrica ZNET	216
Figura 77: Clave AES	216
Figura 78: Parámetros de seguridad configurados	217
Figura 79: Asistente agregar dispositivo Bluetooth	218
Figura 80: Configuración Bluetooth	218
Figura 81: Selección de dispositivo Bluetooth	219
Figura 82: Configuración de seguridad Bluetooth	220
Figura 83: Selección de servicios Bluetooth	220
Figura 84: Finalización del asistente de configuración Bluetooth	221
Figura 85: Conexión módulo Zigbee y microcontrolador	222
Figura 86: Diagrama circuital Multiplexor de voltaje	223
Figura 87: Diagrama circuital Detector de energía	224
Figura 88: Diagrama de Simulación Detector de luz	225
Figura 89: Diagrama Circuital Detector de luz	226
Figura 90: Circuito Detector de luz enrutado en ARES	226
Figura 91: Vista 3D Detector de luz	227
Figura 92: Bottom copper (Cobre inferior) Circuito Detector de luz	227
Figura 93: Top silk (Seda superior) Circuito Detector de luz	227
Figura 84: Fotografía del circuito Detector de luz	228
Figura 95: Diagrama de Simulación Detector de gas GLP	229
Figura 96: Diagrama Circuital del Detector de gas GLP	229
Figura 97: Circuito Detector de gas GLP enrutado en ARES	230
Figura 98: Vista 3D Detector de gas GLP	230
Figura 99: Bottom copper (Cobre inferior) Circuito Detector de GAS	231
Figura 100: Top silk (Seda superior) Circuito Detector de GAS	231

Figura 101: Fotografía del circuito Detector de gas GLP _____	231
Figura 102: Diagrama circuital Inversor de giro motor DC _____	233
Figura 103: Circuito inversor de giro de motor DC enrutado en ARES _____	234
Figura 104: Bottom copper (Cobre inferior) Circuito inversor de giro de motor DC _____	234
Figura 105: Top silk (Seda superior) Circuito inversor de giro de motor DC _____	235
Figura 106: Fotografía circuito Inversor de giro motor DC _____	235
Figura 107: Diagrama de Simulación Módulo Xbee1 _____	238
Figura 108: Diagrama Circuital Módulo Xbee1 _____	239
Figura 109: Circuito Módulo Xbee1 enrutado en ARES _____	240
Figura 110: Vista 3D inferior Módulo Xbee1 _____	241
Figura 111: Vista 3D superior Módulo Xbee 1 _____	241
Figura 112: Bottom copper (Cobre inferior) Circuito Módulo Xbee1 _____	242
Figura 113: Top copper (Cobre superior) Circuito Módulo Xbee 1 _____	242
Figura 114: Top silk (Seda superior) Circuito Módulo Xbee 1 _____	243
Figura 115: Fotografía inferior del circuito Módulo Xbee 1 _____	243
Figura 116: Fotografía superior del circuito Módulo Xbee 1 _____	243
Figura 117: Diagrama de Simulación Módulo Xbee 2 _____	246
Figura 118: Diagrama Circuital Módulo Xbee 2 _____	247
Figura 119: Circuito Módulo Xbee 2 enrutado en ARES _____	248
Figura 120: Vista 3D inferior Módulo Xbee 2 _____	249
Figura 121: Vista 3D superior Módulo Xbee 2 _____	249
Figura 122: Bottom copper (Cobre inferior) Circuito Módulo Xbee 2 _____	250
Figura 123: Top copper (Cobre superior) Circuito Módulo Xbee 2 _____	250
Figura 124: Top silk (Seda superior) Circuito Módulo Xbee 2 _____	251
Figura 125: Fotografía superior del circuito Módulo Xbee 2 _____	251
Figura 126: Diagrama de Simulación Módulo Xbee 3 _____	254
Figura 127: Diagrama Circuital Módulo Xbee 3 _____	255
Figura 128: Circuito Módulo Xbee 3 enrutado en ARES _____	256
Figura 129: Vista 3D inferior Módulo Xbee 3 _____	257
Figura 130: Vista 3D superior Módulo Xbee 3 _____	257
Figura 131: Bottom copper (Cobre inferior) Circuito Módulo Xbee 3 _____	258
Figura 132: Top copper (Cobre superior) Circuito Módulo Xbee 3 _____	258
Figura 133: Top silk (Seda superior) Circuito Módulo Xbee 3 _____	259
Figura 134: Fotografía inferior del circuito Módulo Xbee 3 _____	259
Figura 135: Fotografía superior del circuito Módulo Xbee 3 _____	260
Figura 136: Diagrama de Simulación Módulo GLCD _____	262
Figura 137: Diagrama Circuital Módulo GLCD _____	264
Figura 138: Circuito Módulo Xbee 3 enrutado en ARES _____	264
Figura 139: Vista 3D inferior Módulo GLCD _____	265



Figura 140: Vista 3D superior Módulo GLCD _____	265
Figura 141: Bottom copper (Cobre inferior) Circuito Módulo GLCD _____	266
Figura 142: Top copper (Cobre superior) Circuito Módulo GLCD _____	266
Figura 143: Top silk (Seda superior) Circuito Módulo GLCD _____	267
Figura 144: Fotografía inferior del circuito Módulo GLCD _____	267
Figura 145: Fotografía superior del circuito Módulo GLCD _____	268
Figura 146: Diagrama circuital interfaz Xbee Serial Explorer V12 USB _____	269
Figura 147: Vista superior Xbee Serial Explorer V12 USB _____	270
Figura 148: Vista inferior Xbee Serial Explorer V12 USB _____	270
Figura 149: Fotografía del Xbee Serial Explorer V12 USB _____	271
Figura 150: Fotografía del Xbee Serial Explorer V12 USB Xbee series 2 _____	271
Figura 151: Pasos para abrir GLCD Bitmap editor _____	286
Figura 152: GLCD Bitmap generator _____	287
Figura 153: Coordenadas de la imagen Tipo de Usuario _____	290
Figura 154: Coordenadas de la imagen Ingresar clave _____	291
Figura 155: Coordenadas de la imagen control de la iluminación _____	292
Figura 156: Coordenadas de la imagen Monitoreo de sensores _____	294
Figura 157: Coordenadas de la imagen Control de cerraduras _____	295
Figura 158: Estadísticas de la memoria RAM Y ROM _____	297
Figura 159: Simulación del módulo GLCD en Proteus _____	298
Figura 160: Fotografía Pantalla Iniciando _____	299
Figura 161: Fotografía Pantalla ingresar tipo de usuario _____	299
Figura 162: Fotografía Pantalla ingresar código del usuario _____	299
Figura 163: Fotografía Pantalla control de iluminación. _____	300
Figura 164: Fotografía monitoreo de sensores. _____	300
Figura 165: Fotografía Pantalla Control de cerraduras _____	300
Figura 166: Fotografía Pantalla Bloqueo excedió el número de intentos _____	301
Figura 167: Desarrollador Mobile Processing 007 _____	301
Figura 168: Fondo bg.png _____	302
Figura 169: Fondo us.png _____	302
Figura 170: Fondoilu.png _____	303
Figura 171: Fotografía Pantalla Bienvenida _____	304
Figura 172: Fotografía Pantalla Inicio de sesión _____	305
Figura 173: Fotografía Pantalla control iluminación y monitoreo. _____	308
Figura 174: Como agregar librerías en Netbeans Paso 1 _____	336
Figura 175: Como agregar librerías en Netbeans Paso 2 _____	336
Figura 176: Como agregar librerías en NetBeans Paso 3 _____	337
Figura 177: Como agregar librerías en NetBeans Paso 4 _____	337
Figura 178: Cámara IP Mostrada en el servidor Web _____	341

Figura 179: Interfaz de inicio de sesión	342
Figura 180: Interfaz Usuario Incorrecto	343
Figura 181: Interfaz Usuario Incorrecto	344
Figura 182: Interfaz Cerrar sesión	345
Figura 183: Menú tipo Invitado	345
Figura 184: Menú tipo Administrador	345
Figura 185: Páginas de navegación o Flujo de Página	346
Figura 186: Interfaz Monitorear	348
Figura 187: Interfaz Control de Iluminación	350
Figura 188: Interfaz Control de Cerraduras	352
Figura 189: Interfaz Control de Cortina y Bocina	354
Figura 190: Interfaz Control Modos	355
Figura 191: Interfaz Iniciar Servicios	360
Figura 192: Interfaz Notificaciones SMS	362
Figura 193: Interfaz Notificaciones Correo	365
Figura 194: Interfaz Notificaciones Guardar Historial	367
Figura 195: Consultar Usuario por medio de Ajax Paso 1	369
Figura 196: Consultar Usuario por medio de Ajax Paso 2	370
Figura 197: Consultar Usuario por medio de Ajax Paso 3	370
Figura 198: Consultar Usuario por medio de Ajax Paso 4	370
Figura 199: Consultar Usuario por medio de Ajax Paso 5	371
Figura 200: Consultar Usuario por medio de Ajax Paso 6	371
Figura 201: Configuración de los componentes mediante Ajax	372
Figura 202: Consultar Usuario por medio de Ajax Paso 7	372
Figura 203: Consultar Usuario por medio de Ajax Paso 8	373
Figura 204: Conexión con la base de datos	374
Figura 205: Insertar tabla en la interfaz Usuarios	374
Figura 206: Editar sentencia SQL	374
Figura 207: Interfaz donde se almacena la consulta SQL	375
Figura 208: Agregar librerías mediante el Fix Imports	376
Figura 209: Agregar librerías Logger	377
Figura 210: Agregar librerías Logger	378
Figura 211: Interfaz final de la consulta de usuario	378
Figura 212: Interfaz Ingreso de Usuarios	379
Figura 213: Interfaz eliminar Usuario	381
Figura 214: Interfaz eliminar Usuario	381
Figura 215: Interfaz Gestión Base de Datos	383
Figura 216: Interfaz Tabla histo_modos	384
Figura 217: Interfaz Tabla histo_xbee	385

Figura 218: Interfaz Tabla historial	386
Figura 219: Interfaz Tabla medio	388
Figura 220: Interfaz Tabla modos	389
Figura 221: Interfaz Tabla puntos_finales	391
Figura 222: Interfaz Tabla sensores_actuadores	392
Figura 223: Interfaz Tabla tipo	394
Figura 224: Interfaz Tabla xbee	395
Figura 225: Tabla Usuarios sin encriptación	397
Figura 226: Tabla Usuarios con encriptación	398
Figura 227: Consulta a la tabla usuarios desencryptación	398
Figura 228: Prueba Compatibilidad Nokia e63 y Navegador Opera Mobile 10	400
Figura 229: Prueba Compatibilidad Nokia e63 Ingreso al sistema	400
Figura 230: Prueba Compatibilidad Nokia e63 Control de Iluminacion	400
Figura 231: Test a 1 metro del coordinador	402
Figura 232: Test a 10 metros del coordinador	402
Figura 233: Test a 50 metros del coordinador	403
Figura 234: Test a 75 metros del coordinador	403
Figura 235: Test a 85 metros del coordinador	403
Figura 236: Test a 95 metros del coordinador	403
Figura 237: Test a 100 metros del coordinador	404
Figura 238: Test en el cuarto del servidor	405
Figura 239: Test en el Dormitorio 1	405
Figura 240: Test en la Sala	405
Figura 241: Test en la cocina	405
Figura 242: Test en el Exterior	406
Figura 243: Prueba Red MESH	407
Figura 244: Hyperterminal Bluetooth	408
Figura 245: Pruebas funcionamiento Módulo Xbee 1	409
Figura 246: Pruebas funcionamiento Módulo Xbee 2	409
Figura 247: Pruebas funcionamiento Módulo Xbee 3	410
Figura 248: Pruebas funcionamiento GLCD	411

## ÍNDICE DE TABLAS

Tabla 1: Ventajas y desventajas Tecnologías Inalámbricas	72
Tabla 2: Ventajas y desventajas Servidores de Bases de Datos	77
Tabla 3: Ventajas y Desventajas Lenguajes de Programación	83

Tabla 4: Ventajas y desventajas Servidores Web.....	86
Tabla 5: Ventajas y desventajas servidores de Correo Web.....	89
Tabla 6: Diferencias entre Módulos Xbee Serie 1 y 2.....	100
Tabla 7: Comparación de tipos de UPS .....	117
Tabla 8: Nomenclatura para los módulos de comunicaciones .....	126
Tabla 9: Nomenclatura para el tipo de dispositivo final.....	126
Tabla 10: Nomenclatura dispositivo final luces.....	127
Tabla 11: Nomenclatura dispositivo final detectores de movimiento .....	127
Tabla 12: Nomenclatura dispositivo final detector de humo .....	127
Tabla 13: Nomenclatura dispositivo final Cerraduras eléctricas.....	127
Tabla 14: Nomenclatura dispositivo final Sensores magnéticos .....	127
Tabla 15: Nomenclatura dispositivo final detector de GAS GLP.....	128
Tabla 16: Nomenclatura dispositivo final sensor fotovoltaico.....	128
Tabla 17: Nomenclatura dispositivo final Bocina.....	128
Tabla 18: Nomenclatura estado del dispositivo final .....	128
Tabla 19: Códigos de comunicación Xbee 1 enviados hacia el servidor .....	130
Tabla 20: Códigos de comunicación Xbee 2 enviados hacia el servidor .....	130
Tabla 21: Códigos de comunicación Xbee 3 enviados hacia el servidor .....	131
Tabla 22: Códigos de comunicación Control de usuario, enviados hacia el servidor .....	131
Tabla 23: Códigos de comunicación Gestión de la iluminación enviados hacia el servidor .....	132
Tabla 24: Códigos de comunicación Monitoreo de sensores enviados hacia el servidor.....	132
Tabla 25: Códigos de comunicación Control de Cerraduras enviados hacia el servidor .....	132
Tabla 26: Códigos de comunicación Xbee 1 enviados desde el servidor .....	133
Tabla 27: Códigos de comunicación Xbee 2 enviados desde el servidor .....	133
Tabla 28: Códigos de comunicación Xbee 3 enviados desde el servidor .....	133
Tabla 29: Códigos de comunicación GLCD enviados desde el servidor.....	134
Tabla 31: Códigos de comunicación Gestión de iluminación Bluetooth enviados hacia el servidor .....	136
Tabla 32: Códigos de comunicación Monitoreo Bluetooth enviados hacia el servidor .....	137
Tabla 33: Códigos de comunicación Control de usuario Bluetooth enviados desde el servidor .....	137
Tabla 34: Códigos de comunicación Monitoreo Bluetooth enviados desde el servidor.....	137
Tabla 35: Códigos de comunicación Gestión de iluminación SMS enviados hacia el servidor .....	139
Tabla 36: Códigos de comunicación Cerraduras SMS enviados hacia el servidor.....	139
Tabla 37: Códigos de comunicación Monitorear estados SMS enviados hacia el servidor .....	140
Tabla 38: Códigos de comunicación Ayuda SMS enviados hacia el servidor .....	140
Tabla 39: Códigos de comunicación Monitoreo de iluminación SMS enviados desde el servidor ..	141
Tabla 40: Códigos de comunicación Monitoreo de sensores SMS enviados desde el servidor.....	141
Tabla 41: Códigos de comunicación Ayuda SMS enviados desde el servidor.....	141

# **CAPÍTULO 1**

## **ANTECEDENTES**

Este capítulo hace referencia al porque del proyecto de Tesis, aquí se encontrará el objetivo general que se persigue, previamente estableciendo cual es la solución que se dará al problema planteado, mediante una justificación clara y precisa, y así establecer los alcances de la tesis.

### **1.1. PROBLEMA**

La creciente inseguridad y la necesidad de mayor confort, obligan a buscar nuevos e innovadores métodos de protección y comodidad, la constante preocupación de no tener vigilada y controlada una residencia en todo momento y lugar, sin saber con certeza que es lo que ocurre en su interior, teniendo presente que pueda ocurrir cualquier eventualidad como un incendio u otro incidente doméstico.

Diversas personas sufren de alguna discapacidad física que impide su libre movilidad dentro de su residencia, llevándolos a valerse de otras personas para que realicen tareas rutinarias tales como: gestionar la iluminación (en dormitorios, cocina y demás partes de la residencia), abrir o cerrar cortinas y cerraduras.

### **1.2. OBJETIVOS**

#### **1.2.1. GENERAL**

Diseñar una red Mesh/Zigbee gestionada mediante SMS (Short Message Service), servidor web, correo electrónico, aplicación bluetooth para dispositivos móviles y control inalámbrico local que permita controlar una residencia.

### **1.2.2. ESPECÍFICOS**

- Gestionar iluminación, cerraduras, persianas, alarmas, sensores de movimiento, sensores de humo y cámaras de seguridad por medio de dispositivos móviles tales como: celulares, PDA, notebooks, etc.
- Recibir notificaciones permanentes del comportamiento de los sensores y dispositivos implementados en una residencia de forma inmediata al celular y correo electrónico.
- Almacenar en una base de datos el comportamiento de la red para conocer hora y fecha en el que ocurren eventos.
- Implementar un Simulador de presencia que sea capaz de encender o apagar luces, abrir o cerrar cortinas cuando la residencia este vacía.
- Integrar sistemas domóticos y de telecomunicaciones por medio de redes Mesh/Zigbee.

### **1.3. JUSTIFICACIÓN**

- Uso racional de la energía eléctrica, al disminuir el tiempo en que la iluminación permanece activa en una residencia, por ejemplo cuando no se detecta presencia en su interior o la luz natural hace innecesario su uso.
- Prevención de tentativa de robo o hurto, al usar el modo de presencia se podría hacer desistir al ladrón de violentar o atentar contra la seguridad de una residencia.
- Identificación de tentativa de robo o hurto, cuando el simulador de presencia no cumplió con su objetivo y las cerraduras se ven forzadas o se detecta movimiento dentro de una residencia, de manera inmediata el sistema notifica al usuario por medio de SMS y/o correo electrónico el evento ocurrido, también visualiza en tiempo real mediante las cámaras de seguridad la residencia por medio del internet.
- Confort del usuario al facilitar tareas comunes, especialmente si sufre de alguna discapacidad física, con solo presionar un botón toma control de la residencia.

- Detección de posible incendio, al activarse los sensores de humo o gas se notificara por medio de SMS y/o correo electrónico, también es posible visualizar el suceso en tiempo real mediante las cámaras de seguridad para tener la certeza de lo que ocurre y tomar las medidas necesarias.
- Zigbee es un nuevo estándar que facilita la comunicación en sistemas domóticos y proporciona mayor cobertura, menor costo y menor consumo de energía, la cobertura se ve ampliada al implementar una red Mesh o malla que permiten a todos los dispositivos Zigbee comunicarse entre ellos, es así que al estar al otro extremo del nodo central igual se tiene conectividad.
- Control vía web, para personas que viajan continuamente y necesitan saber el estado de una residencia a distancia, pueden gestionar todo desde cualquier lugar del mundo.
- Optar por el uso de software libre en la mayoría de nuestras aplicaciones para reducir costos y problemas referentes al uso de licencias.

#### **1.4. ALCANCE**

El sistema será implementado en su totalidad en el hogar del Sr. Miguel Ángel Cedeño Villarroel cuya dirección es: OE3F Malvas BQ-33 S29-15 PISO 4 DEP 403 S29 Borbón Sector Turubamba Alto.

**Nota:** La ubicación del sitio de implementación se observa en el Anexo 1.

Los dispositivos sensores y actuadores estarán comunicados a la caja central, por medio de un red Mesh/Zigbee para evitar el excesivo cableado que daña la decoración de la residencia, mediante ésta red se mejora la seguridad y el confort, gracias al control remoto inalámbrico que estará unido y configurado a la red Zigbee, de modo que no será necesario estar cerca del receptor central sino solo a uno de los nodos de la red Mesh.

El sistema está comprendido por los siguientes servicios:

#### **1.4.1. SEGURIDAD FÍSICA**

- Alarmas técnicas (humo o fuego).
- Alarmas de Intrusión (control de movimiento, cámaras de seguridad, bocina).
- Prevención de fallo de suministro eléctrico mediante el uso de UPS conectados al servidor principal y caja de circuitos.
- Notificación de eventualidades mediante SMS y/o correo electrónico, en el acto o cada tiempo determinado por el usuario.
- Uso de Zigbee para la comunicación del sistema con los respectivos sensores y actuadores, no de cables porque de esta manera se evitará su corte ya sea accidental e intencionalmente.
- Control de sistema de seguridad vía aplicación web, SMS, Dispositivo móvil con bluetooth y control remoto local táctil.

#### **1.4.2. SEGURIDAD LÓGICA**

##### **1.4.2.1. En la aplicación Web**

- Control de usuarios mediante el uso de contraseñas almacenadas en una base de datos.
- Uso de Antivirus y Antispyware.

##### **1.4.2.2. En la aplicación bluetooth**

- Clave de ingreso para el uso de la aplicación (almacenada en la base de datos).
- Emparejamiento previo de dispositivos bluetooth con clave.



#### **1.4.2.3. SMS (Short Message Service)**

- Control de usuario, cuando se requiera abrir una cerradura, mediante SMS enviar el mensaje de texto con el comando y anteponiendo la contraseña.

#### **1.4.2.4. Confort**

- Control de iluminación (on/off).
- Control de cortinas eléctricas.
- Control de sistema vía aplicación web, SMS, Dispositivo móvil con bluetooth y control remoto inalámbrico local táctil.

#### **1.4.2.5. Simulación de presencia**

- Es un servicio que ofrece el sistema domótico capaz de simular presencia en el interior de una residencia, consiste en activar o desactivar interruptores para hacer que la iluminación cambie, también la apertura y cierre de cortinas en un lapso de tiempo determinado previamente.
- Activación del control de presencia mediante aplicación web.

#### **1.4.2.6. Modo ahorro de energía**

- Al accionar este modo los focos encendidos se apagan al no detectar movimiento en la habitación y se procede a la apertura de cortinas en caso de haber luz natural.
- Los focos exteriores se encenderán solo cuando sea de noche y/o haya movimiento.

#### **1.4.2.7. Bases de Datos**

- Se contara con un redoblaje de seguridad de la base de datos en su totalidad, habrá una opción para respaldar completamente los datos de forma mensual o de forma manual, cuando sea requerido.
- De igual manera se realizaran encriptaciones de los campos de las tablas que contengan claves o contraseñas.

#### **1.4.3. HISTÓRICOS**

- Almacenar en una base de datos el comportamiento del sistema para conocer hora y fecha en el que ocurren eventos relevantes en la residencia.
- Consultar todos los eventos pasados cuando se necesite.

## **CAPÍTULO 2**

### **SUSTENTO TEÓRICO**

En este capítulo, se proporciona las bases teóricas que sustentan el proyecto de tesis, tecnologías, dispositivos, módulos utilizados para el posterior análisis, desarrollo e implementación del proyecto.

#### **2.1. DOMÓTICA**

La domótica es la nueva tendencia de hoy en día en los hogares modernos, basándose en su significado la palabra domótica se divide en dos: el latín “domus” que quiere decir “casa” y del sufijo “tica” que se encuentra también en las palabras telemática, robótica, informática; y hace referencia a la palabra “automática” en conclusión “gestión por medios informáticos”.

Para comprender cuál es la finalidad de la creación de éste proyecto de tesis, es necesario centrarse en los beneficios que ofrece la domótica, que convierte un hogar común en un sitio inteligente, confortable, agradable y seguro.

##### **2.1.1. PILARES FUNDAMENTALES**

Los 4 pilares básicos en los que se fundamenta la domótica son:

###### **2.1.1.1. Seguridad de bienes y personas**

- Detección de posibles intrusos por medio del uso de sensores.
- Simulación de presencia.
- Alarmas ante incendios, fugas de gas, escapes de agua etc.
- Apertura y cierre de cortinas o persianas de forma programada.
- Monitoreo mediante cámaras de vigilancia.

#### **2.1.1.2. Gestión y ahorro de energía**

- Programación y monitoreo de encendido/apagado de luces o sistemas de calefacción.
- Desconexión de equipos o reducción de la potencia de energía.
- Gestión de tarifas a pagar del consumo eléctrico, mediante el control de los dispositivos manejados.

#### **2.1.1.3. Confort y entretenimiento dentro del hogar**

- Apagado y encendido de luces del hogar.
- Automatización del apagado/ encendido de luces.
- Regulación de la iluminación.
- Automatización de equipos electrónicos.

#### **2.1.1.4. Comunicación y trabajo**

- Controles remotos para manejo de dispositivos.
- Comunicación de alarmas hacia dispositivos móviles.

### **2.1.2. ARQUITECTURAS DE UN SISTEMA DOMÓTICO**

Al realizar un sistema domótico se toma en cuenta la ubicación, espacio, características, distribución de los distintos dispositivos que serán parte del sistema, basándose en éstos puntos se define tres tipos de arquitectura:

#### **2.1.2.1. Arquitectura centralizada**

En esta arquitectura existe un módulo central que gestiona todos los dispositivos domóticos mediante los mensajes que le envíen, y al analizarlos controla los dispositivos actuadores, la desventaja que se da en esta arquitectura es que cuando el módulo central sufre algún daño el sistema colapsa.

### **2.1.2.2. Arquitectura distribuida**

Aquí todos los módulos son inteligentes es decir, autónomos al realizar sus tareas no requieren que alguien más actúe sobre ellos, pueden ser ubicados en cualquier parte del hogar, esta arquitectura es ideal para sistemas inalámbricos y de cableado tipo bus.

### **2.1.2.3. Arquitectura mixta**

Arquitectura puramente inalámbrica en la cual los sensores y actuadores actúan y procesan información que la envían al resto de dispositivos que forman parte del sistema es la composición de las arquitecturas anteriores.

## **2.2. TECNOLOGÍA ZIGBEE**

Zigbee es un protocolo de comunicación inalámbrica reconocido a nivel mundial creado por la ZIGBEE ALLIANCE<sup>1</sup> organismo sin fines de lucro formada por más de 200 empresas<sup>2</sup> repartidas en el mundo, basado en el estándar IEEE802.15.4<sup>3</sup> ideal para redes domóticas con aplicaciones que requieren comunicaciones seguras con bajas tasas de envío-recepción de datos y alta vida útil de sus baterías.

---

<sup>1</sup> "La Zigbee Alliance es una asociación de empresas que trabajan conjuntamente de manera fiable, rentable, de bajo consumo, interconectados mediante red inalámbrica, de monitoreo y control basados en un estándar global abierto." (<http://www.zigbee.org/About/OurMission.aspx>).

<sup>2</sup> Entre las 200 empresas algunas de las más reconocidas: Motorola, Huawei Technologies, Samsung Electronics, ETRI, KDDI R&D Laboratories Inc., OKI Electric Industry, Orange FT, Telecom Italia y TSC Systems.

<sup>3</sup> "IEEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos." ([http://es.wikipedia.org/wiki/IEEE\\_802.15.4](http://es.wikipedia.org/wiki/IEEE_802.15.4)).

En ambientes domóticos esta tecnología facilita el control de iluminación, la seguridad del domicilio, gestión y administración de dispositivos electrónicos que manejan un bajo consumo eléctrico dentro de ambientes industriales y especialmente domóticos, eliminando el excesivo y molesto cableado que regularmente se ve en una casa o pequeña oficina, todo esto a un precio asequible, convirtiendo un lugar común en un lugar inteligente y seguro.

Zigbee realiza su comunicación en la banda libre 2.4 GHz y a diferencia de otras tecnologías su comunicación se realiza mediante un único canal.

Cabe destacar que una red Zigbee contiene un máximo de hasta 65535 equipos conectados teóricamente, pero prácticamente ésta cantidad es menor siendo de igual forma miles de dispositivos Zigbee que pueden ser manejados por éste protocolo.

### **2.2.1. ARQUITECTURA DE UNA RED ZIGBEE**

Zigbee es una arquitectura basada en el modelo OSI<sup>4</sup> definiendo sus propias capas de la siguiente manera:

Las dos capas más bajas son definidas por el estándar IEEE 802.15.4: la Capa Física y la Subcapa de Control de Acceso al Medio (MAC<sup>5</sup>), que son utilizadas por Zigbee para establecer un entorno de trabajo para sus aplicaciones.

---

<sup>4</sup> Modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) define 7 capas que son: Física, Enlace de Datos, Red, Transporte, Sesión, Presentación, Aplicación.

<sup>5</sup> MAC (Media Access Control)

### **2.2.1.1. Capas de una Red Zigbee**

En la arquitectura Zigbee se tiene las siguientes capas:

#### **2.2.1.1.1. Capa Física**

Trabaja en cualquiera de éstos rangos de frecuencias: 868 MHz, 915 MHz o 2.4 GHz alcanzando velocidades de 20 Kbps, 40 Kbps y 250 Kbps, permitiendo alcanzar distancias de un máximo 100 metros con un muy bajo consumo de energía haciéndola ideal para aplicaciones domóticas.

#### **2.2.1.1.2. Capa de MAC**

Proporciona servicios que permiten la fiabilidad y la comunicación directa entre dispositivos.

Zigbee aporta con más capas al estándar IEEE 802.15.4 las cuales son:

#### **2.2.1.1.3. Capa de red NWK (Network Layer)**

Diseñada para gestionar el enrutamiento y mantenimiento de los demás nodos que son parte de la red.

#### **2.2.1.1.4. Capa Aplicación (APL Application Layer)**

Se subdivide en la subcapa APS (Application Support Layer) que tiene como función dar una interfaz a la capa red y la subcapa ZDO (Zigbee Device Objects) encargada de inicializar la subcapa APS y a la subcapa NWK.

De ésta manera Zigbee conjuntamente con la IEEE 802.15.4 conforman una arquitectura completa de protocolos la cual permite la comunicación de una gran cantidad de dispositivos o nodos dentro de una misma red. Todas las transmisiones que se realizan mediante Zigbee en la banda 2.4 GHz especifican 16 canales (11 al 26) de los cuales solo se ocupará uno, el cual es capaz de tener actividad dentro de otras tecnologías. Para escoger el canal apropiado para trabajar se evalúa el nivel de la señal mediante 2 procedimientos:

ED (Energy Detection) y CCA (Clear Channel Assessment), los que evalúan una señal compatible con 802.15.4 y el nivel de energía presente en el canal.

Zigbee al estar en una banda libre permite la coexistencia con otras frecuencias similares como 802.11 b - g y proveen alternativas para sitios donde existe una alta proliferación de redes WiFi.

## **2.2.2. ELEMENTOS DE UNA RED ZIGBEE**

Una red Zigbee está conformada por 3 elementos básicos los cuales se describe a continuación:

### **2.2.2.1. Dispositivo Coordinador**

Elemento central y único responsable de formar la red, establecer el canal de comunicación, requisitos de seguridad y de escoger el PAN ID (identificador único de 16-bit, escogido en el momento de la creación de la red), por definición debe haber solo un Coordinador por red.

Cuando el dispositivo coordinador ha logrado establecer éstos parámetros iniciales está en la capacidad de comunicarse con los dispositivos Router y End Device para realizar las funciones de un Router dentro de la red, siendo el encargado de enrutar los paquetes y ser origen o destinatario de la información que se manejará en la red.



#### **2.2.2.2. Router**

La primera tarea de éste elemento es unirse a la red Zigbee creada por el dispositivo coordinador para que pueda tener la facultad de interactuar con él y los demás dispositivos Router y End Device que se encuentren en la red.

El Router es capaz de crear y guardar información de la red para establecer la mejor ruta por la cual será enviado el paquete de información, éste dispositivo está configurado para siempre estar en modo de recepción o de escucha para cuando sea requerido por los demás Routers cercanos a él.

#### **2.2.2.3. End Device**

Los dispositivos finales no tienen la capacidad de enrutar paquetes siempre tienen que depender de su nodo padre, que puede ser un Coordinador o Router motivo por el cual no envían información a otros End Device directamente, al no tener funciones de enrutamiento tienen un menor consumo de energía y optimizan la vida de sus baterías.

La comunicación entre los End Device y los Router padres se realiza mediante una encuesta que envían los dispositivos finales para hacerle conocer al nodo padre que se encuentra despierto y dispuesto a recibir información, por la otra parte el Router padre responde la no existencia de paquetes para ser enviados o si no fuera el caso envía los mensajes disponibles directamente al End Device.

### **2.2.3. TOPOLOGÍAS DE RED ZIGBEE**

#### **2.2.3.1. Star o Estrella**

Esta topología tiene una larga vida útil consecuentemente por el bajo consumo que requiere.

#### **2.2.3.2. Mesh o Malla**

Con ésta topología se tiene una gran fiabilidad al contar con varias opciones de enrutamiento para alcanzar el destino final.

#### **2.2.3.3. Clúster Tree o Racimo de Árbol**

Topología en la cual se complementan la topología del tipo Mesh y la topología Estrella obteniendo los beneficios de ambas topologías.

### **2.2.4. ESTRATEGIAS DE CONEXIÓN EN UNA RED ZIGBEE**

Los nodos “esclavos” que son parte de una red Zigbee tienen la capacidad de permanecer por mucho tiempo en modo “dormido”, debido a que si no se requiere de él no hace falta que esté consumiendo energía innecesariamente, es así que la estrategia que utilizan es la de permanecer en estado pasivo o “dormido” hasta el momento que se hace necesario su uso o cuando el dispositivo “coordinador” desea saber el estado de los demás dispositivos de la red, de esta forma los nodos “esclavos” solo “despertarán” por un mínimo lapso de tiempo para informar su estado y terminado esto vuelven a sus estado pasivo, el tiempo que se toman los nodos en este cambio de estado es de 15ms.

#### **2.2.4.1. Con balizas**

Esta estrategia se utiliza para sincronizar los dispositivos de la red en determinados intervalos de tiempo, asignados por el Dispositivo Coordinador de Red que pueden ser desde los 15ms hasta 4 minutos, la estrategia con balizas es utilizada cuando el Coordinador es alimentado por una batería. Al realizar el “balizamiento” es decir, el envío de mensajes en Broadcast a los demás dispositivos de la red para conocer si alguno de ellos desea intervenir primero los dispositivos End Device deberán registrarse ante el Coordinador para conocer si existe mensajes para él, caso contrario esperar a que se termine el “balizamiento” y volver al estado pasivo o “dormido” hasta que nuevamente el Coordinador realice el “balizamiento” en el tiempo que estableció previamente.

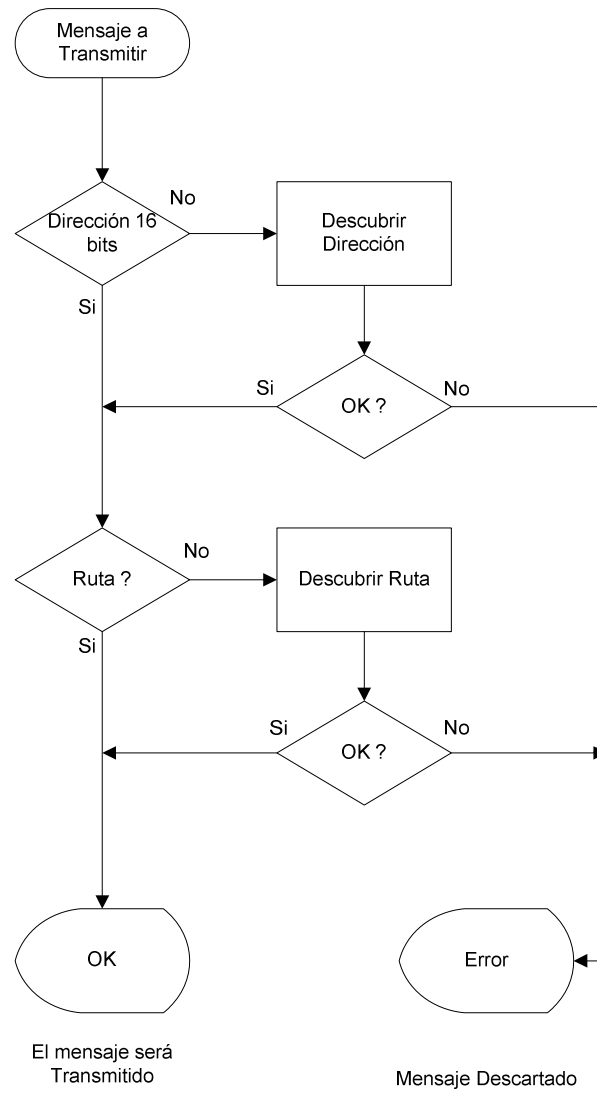
#### **2.2.4.2. Sin balizas**

En la estrategia sin balizas los dispositivos actúan de forma autónoma, es por esto que cualquiera de ellos puede iniciar una conversación hacia el dispositivo Coordinador cuando crea necesario u ocurra algún evento que el Coordinado tenga que conocer.

Esta estrategia es ideal para sistemas de seguridad donde se maneja sensores, detectores, etc., debido a que en la totalidad de tiempo los dispositivos estarán en estado pasivo y únicamente “despiertan” cuando se produzca algún evento o haga conocer a la red que se encuentra “vivo”, en cualquiera de éstos casos el Coordinador recibe el mensaje y activa las alarmas correspondientes. La estrategia sin balizas se requiere que el dispositivo Coordinador de red esté permanentemente conectado a alguna fuente de energía.

En el siguiente diagrama de flujo se explicara en resumen como realiza la transmisión de mensajes por medio de Xbee.

En la siguiente figura, se muestran el diagrama de flujo del envío de un mensaje en una red Zigbee.



**Figura 1: Diagrama de Flujo Transmisión Mediante Xbee**

## 2.3. MÓDULOS DE COMUNICACIONES

### 2.3.1. MÓDULOS GSM

GSM Global System for Mobile communications (Sistema Global para las comunicaciones Móviles), sistema que provee un estándar común para los usuarios permitiendo a una red digital de teléfonos móviles el envío de mensajes de texto (SMS), archivos de imagen, soporte de voz-video, Roaming<sup>6</sup> internacional, navegar por Internet y otros servicios de valor agregado. Es así que GSM se convierte en un sistema de enorme utilidad en el mundo, con el cual se accede a todo un sin número de servicios convirtiendo a un teléfono en una central de información, en donde se consulta información meteorológica, deportiva, actualidad, cotización de monedas extranjera, enviar y recibir correo electrónico desde cualquier lugar en que se encuentre, sin necesidad de disponer de un computador a todo momento.

#### 2.3.1.1. GSM - SIM Card

SIM (Módulo de Identidad del Abonado), mediante ésta tarjeta los usuarios cambian de dispositivo GSM fácilmente sin tener la necesidad de configurar el nuevo dispositivo ni la pérdida de servicios de suscripción personalizados tales como mensajería. De igual forma ésta tarjeta SIM hace posible para el usuario migrar de operador GSM y mantener el mismo teléfono.

Es por esto que la tecnología GSM se basa en el uso de la tarjeta SIM en la cual se almacena todos los datos del usuario cabe citar que los servicios están asociados a la tarjeta SIM y no al equipo móvil.

---

<sup>6</sup> ROAMING es la capacidad de los dispositivos móviles para moverse desde una zona de cobertura hacia otra, sin pérdida de la conectividad y de ésta manera realizar y recibir llamadas en redes móviles fuera del área de servicio local de su compañía o durante una estancia en otro país diferente, con la red de una empresa extranjera.

### **2.3.1.2. Seguridad GSM**

GSM ofrece seguridad y confiabilidad a los usuarios del servicio, esto se realiza mediante la Tarjeta SIM en la cual son asignados dos códigos de seguridad PIN y PUK, los cuales son verificados por la red y así pueden ser o no autorizados para el uso del servicio.

#### **2.3.1.2.1. Código PIN**

Código de cuatro cifras que evita que la línea telefónica sea utilizada por personas no autorizadas. El usuario deberá introducir éste código cuando desee conectarse por primera vez a la red y cada vez que encienda su móvil (depende de la configuración realizada por el usuario al dispositivo GSM).

#### **2.3.1.2.2. Código PUK**

Es un código de ocho cifras que permite el desbloqueo del código PIN.

### **2.3.1.3. Comandos AT con GSM**

Cualquier dispositivo GSM (depende del dispositivo móvil) puede ser conectado a través de un computador para enviar y recibir mensajes por e-mail, faxes, navegar por Internet, tener acceso seguro a cualquier red, de igual forma se utilizan otras funciones digitales de transmisión de datos, como Servicio de Mensajes Cortos (SMS) que se manipulan desde el mismo computador, pudiendo leer los mensajes recibidos y almacenados, enviar mensajes a otros usuarios, realizar llamadas, recibir llamadas y todas las funciones que posea el dispositivo móvil GSM, esto se realiza gracias a la ayuda de los comandos AT<sup>7</sup>.

---

<sup>7</sup> Los comandos AT (Attention) son instrucciones codificadas que conforman un lenguaje de comunicación entre el usuario Serial y un MODEM. Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así configurarlo y proporcionarle instrucciones.

La implementación de los comandos AT corresponde a los dispositivos GSM y no depende del canal de comunicación a través del cual los comandos son enviados, ya sea cable de serie, canal Infrarrojos, Bluetooth, etc.

Entre los comandos AT más utilizados para el envío de SMS se menciona los siguientes:

- AT+CPMS: Seleccionar lugar de almacenamiento de los SMS
- AT+CMGF: Seleccionar formato de los mensajes SMS
- AT+CMGR: Leer un mensaje SMS almacenado
- AT+CMGL: Listar los mensajes almacenados
- AT+CMGS: Enviar mensaje SMS
- AT+CMGW: Almacenar mensaje en memoria
- AT+CMSS: Enviar mensaje almacenado
- AT+CSCA: Establecer el Centro de mensajes a usar
- AT+WMSC: Modificar el estado de un mensaje

**Nota:** Para ver una lista completa de comandos AT ver en el Anexo 2.

### 2.3.2. MÓDULOS XBEE

Son módulos de radio frecuencia diseñados para operar bajo el protocolo Zigbee, cuando se requiera una comunicación segura entre dispositivos remotos con un bajo consumo de energía, son utilizados en su mayoría en aplicaciones de automatización de casas (domótica), sistemas de seguridad, monitoreo de sistemas remotos, aparatos domésticos, alarmas contra incendio, etc.

Éstos módulos operan dentro de la banda ISM<sup>8</sup> (Industrial, Scientific and Medical) utilizando la frecuencia de 2.4 GHz.

---

<sup>8</sup> ISM (Industrial, Scientific and Medical) bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industriales, científicas y médicas.

Los módulos Xbee tienen 2 formas de comunicación: Transmisión serial transparente (modo AT) y el modo API<sup>9</sup>.

Los módulos Xbee pueden ser configurados desde cualquier computador utilizando el software X-CTU o también desde el propio microcontrolador. La potencia de transmisión de los módulos Xbee puede ser de 2mW para 100 mts o 60mW para hasta 1500 mts.

Existen dos series dentro de los módulos Xbee que son Serie 1 y la Serie 2 o también llamada 2.5, los módulos Serie 1 y Serie 2 tienen el mismo pin-out<sup>10</sup>, sin embargo, no son compatibles entre ellos porque utilizan distintos chipset y trabajan con protocolos diferentes.

Los módulos Xbee serie 1 están basados en el chipset<sup>11</sup> de Freescale<sup>12</sup> para ser utilizado en redes punto a punto y punto a multipunto, mientras que los módulos de la serie 2 están basados en el chipset de Ember<sup>13</sup> diseñados para ser utilizados en aplicaciones que requieren repetidores o una red Mesh. Tanto los módulos Serie 1 como Serie 2 pueden utilizar los modos AT y API.

---

<sup>9</sup> "API (Application Programming Interface - Interfaz de programación de aplicaciones). Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc."

<sup>10</sup> Pinout es un término anglosajón que, en traducción libre, significa patillaje, o más correctamente asignación de patillaje. Es usado en electrónica para determinar la función de cada pin en un circuito integrado, o bien en un dispositivo electrónico discreto.

<sup>11</sup> El "chipset" es el conjunto (set) de chips que se encargan de controlar determinadas funciones del ordenador, como la forma en que interacciona el microprocesador con la memoria o la caché, o el control de los puertos.

<sup>12</sup> Freescale Semiconductor, Inc. es un fabricante estadounidense de semiconductores. Creado a partir de la división de semiconductores de Motorola en 2004. Freescale se centra en el mercado de los sistemas integrados y las comunicaciones.

<sup>13</sup> Ember Empresa líder proveedora de sensores inalámbricos y tecnologías de control de red, además es un promotor clave y miembro de la Junta de la Zigbee Alliance.



### 2.3.2.1. Tipos de Módulos Xbee

En la siguiente figura, se muestran los distintos tipos de módulos Xbee:

#### 2.3.2.1.1. Módulos Xbee

Los módulos Xbee tienen un alcance en interiores de hasta 30mts y en exteriores el alcance llega hasta 100mts con antena dipolo que se muestra en la siguiente Figura.

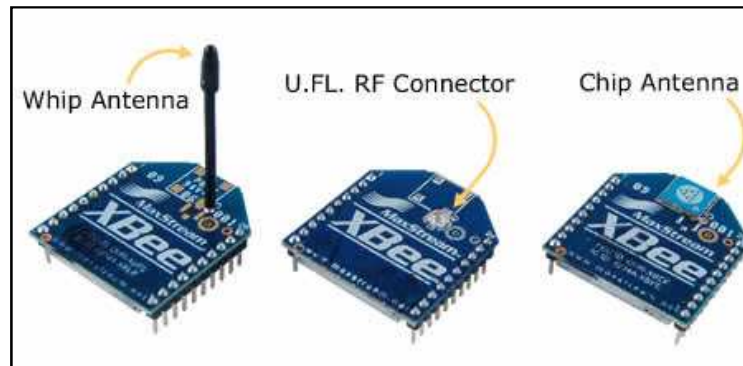


Figura 2: Tipos de Módulos Xbee

Fuente: <http://www.decelectronics.com/html/XBEE/XBEE.htm>

#### 2.3.2.1.2. Módulos Xbee Pro

Los módulos Xbee Pro tienen un alcance en interiores de hasta 100mts y en exteriores el alcance llega hasta 1500mts con antena dipolo que se muestra en la siguiente Figura.

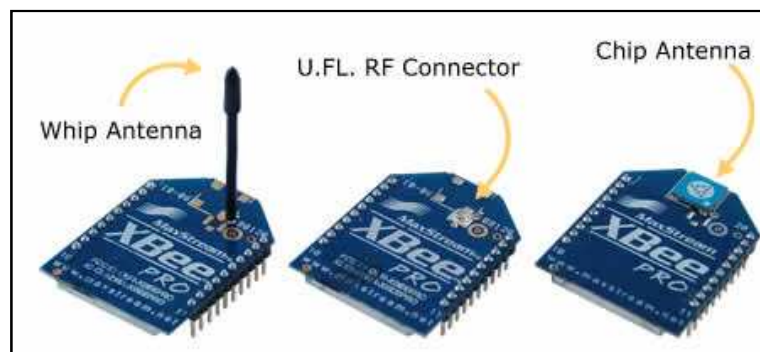


Figura 3: Tipos de Módulos XbeePro

Fuente: <http://www.decelectronics.com/html/XBEE/XBEE.htm>

### **2.3.2.2. Requerimientos de Conexión**

Para la conexión de los módulos Xbee se requiere como mínimo proveer de dos terminales (VCC y GND) y dos de datos (Vin y Vout), para realizar actualizaciones de firmware en los módulos se tiene que agregar las conexiones de los pines RTS y DTR.

Los módulos Xbee no requieren ser soldados porque su diseño les permite ser montados y desmontados de un zócalo que al igual de los módulos deben ser de dos hileras de 10 pines separadas entre ellas por 22mm y una separación entre pines de 2mm.

Los módulos Xbee pueden ser programados a través del Hyperterminal y una interface serial con un Max232<sup>14</sup> y una serie de comandos AT, convirtiéndose en un método muy complicado y tedioso, por éste motivo existen dos tipos de interfaces, serial y USB que con la ayuda del Software X-CTU son utilizadas para programar y definir los parámetros de los módulos Xbee de una manera más rápida.

### **2.3.2.3. Seguridad en los Módulos Xbee**

Proveen un cifrado AES<sup>15</sup> de 128 bits, ésta clave es asignada manualmente y no es posible leerla, solo ingresarla. Además de cifrar la clave, cifra todo lo que se transmite y setea la bandera correspondiente a las tramas de esta manera se transmite tramas seguras. La clave debe ser la misma para todos los dispositivos de la red que intervengan en la comunicación.

Los parámetros que manejan la seguridad son EE (Encryption Enable) y la clave es almacenada en el campo KY.

---

<sup>14</sup> El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL (Tiempo de vida) y viceversa. Lo interesante es que sólo necesita una alimentación de 5V.

<sup>15</sup> AES es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos.

Los módulos Zigbee implementan seguridad de la siguiente manera:

- No poseen un manejo de claves dentro de aplicaciones.
- Poseen una clave de enlace única que se define previamente en el parámetro KY.
- La clave de red única es enviada a todos los componentes de la red desde su Coordinador, esta clave no puede contener el valor 0 y si fuera el caso se asignara aleatoriamente.
- El Coordinador está en la capacidad de configurar el parámetro de seguridad EO (Encryption Options) con el bit 1, haciendo que la clave de red se actualice periódicamente y sea entregada por el Centro de seguridad al momento que algún dispositivo se una a la red.

## **2.4. DISPOSITIVOS ELECTRÓNICOS**

### **2.4.1. MICROCONTROLADORES**

Un microcontrolador es un circuito integrado o chip el cual incorpora la mayor parte de los elementos que configuran un controlador, los Microcontroladores normalmente están compuestos de un Procesador o UCP (Unidad Central de Proceso), memoria y unidades de E/S (Entrada y Salida).

Los Microcontroladores son destinados a satisfacer tareas como: secuenciamiento, codificación/decodificación, monitoreo, adquisición de datos, señalización, procesamiento de señales, control retroalimentado, temporización, cálculos aritméticos sencillos y complejos, comunicaciones, automatización, despliegue digital, control on-off, etc.

### **2.4.1.1. Componentes del microcontrolador**

Un micro controlador está compuesto por:

#### **2.4.1.1.1. El UCP o procesador**

El UCP es el elemento central que determina las principales características a nivel hardware y software de un microcontrolador, tiene como función principal el direccionar la memoria de instrucciones, recibir la instrucción en curso, su decodificación y la ejecución del proceso o tarea que conlleva la instrucción de igual forma se encarga del almacenamiento del resultado.

#### **Arquitectura de los UCP o procesadores:**

Existen 3 arquitecturas las cuales se describen a continuación:

- **CISC:** del inglés Complex Instruction Set Computer, Computadora de Conjunto de Instrucciones Complejo, éste tipo de arquitectura dispone de más de 80 instrucciones que permiten realizar operaciones muy complejas, sofisticadas y potentes entre operandos alojados en los registro internos.
- **RISC:** del inglés Reduced Instruction Set Computer, Computadora con Conjunto de Instrucciones Reducido, estos procesadores cuentan con un reducido número de instrucciones las cuales son simples y se ejecutan en un solo ciclo, esto permite optimizar el hardware y el software del procesador.

Es por esto que los procesadores de los Microcontroladores son de tipo RISC.

- **SISC:** Computadores con Conjunto de Instrucciones Específicas, son Microcontroladores destinados a aplicaciones concretas y específicas es decir, que en ésta arquitectura las instrucciones se adaptan a las necesidades de la aplicación a desarrollar.

#### **2.4.1.1.2. Memoria**

Dentro de los Microcontroladores se encuentran dos tipos de memorias, la memoria de instrucciones y datos, las cuales por un lado una debe ser de tipo ROM o no volátil, destinada a contener el programa de instrucciones y la otra de tipo RAM o volátil que se encarga de almacenar las variables y los datos.

En la memoria ROM solo se almacena un único programa debido a que los Microcontroladores sólo son destinados a realizar una única tarea.

La memoria RAM de estos dispositivos no es de gran capacidad debido a que ésta memoria solo deberá contener las variables y los cambios de información que se produzcan en la ejecución del programa, sin preocuparse de guardar una copia del programa principal pues éste se ejecuta directamente desde la memoria ROM.

Los Microcontroladores pueden tener distintos tipos de memoria ROM a continuación, se describe cinco versiones de memoria no volátil que se puede encontrar dentro de un microcontrolador.

- **ROM con máscara**

Memoria de sólo lectura cuyo contenido es grabado durante la fabricación del chip. Tiene un costo elevado por tal motivo se aconseja su uso únicamente cuando se precisan cantidades superiores a varios miles de unidades.

- **OTP (One Time Programmable)**

Es un tipo de memoria de solo lectura "programable una sola vez". Es el usuario quien escribe el programa en el chip a través de un sencillo grabador controlado por un software desde el computador, éste tipo de memoria se hace ideal en la construcción de prototipos y series pequeñas.

- **EPROM**

La memoria EPROM (Erasable Programmable Read Only Memory) puede borrarse y grabarse muchas veces, la grabación en éste tipo de memoria se realiza de igual forma que en las memorias OTP. Las cápsulas de este tipo de memorias son de material cerámico y son más costosos que los Microcontroladores con memoria OTP que están fabricados con plástico.

- **EEPROM**

Electrical Erasable Programmable Read Only Memory, son memorias de sólo lectura, programables y borrables eléctricamente. La programación y borrado de estas memorias se realizan eléctricamente desde el propio grabador y bajo el control programado de un computador haciendo mucho más fácil realizar modificación en el programa y cuantas veces se requiera.

- **FLASH**

Este tipo de memoria no volátil es de bajo consumo puede ser escrita y borrada 'n' veces, funciona como una ROM y una RAM pero consume menos y es mucho más pequeña.

Esta memoria es aconsejada cuando se requiere una gran cantidad de memoria de programa no volátil, es muy rápida y tolera más ciclos de escritura/borrado.

Tanto la memoria EEPROM como FLASH son muy útiles al permitir que los Microcontroladores puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta, esta reprogramación del microcontrolador se convierte en un trabajo de rutina mientras se pone a punto el programa en construcción.

#### **2.4.1.1.3. Puertas o Líneas de Entrada y Salida**

La mayor utilidad de un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos externos destinados a proporcionar soporte a las señales de entrada, salida y control.

Estas líneas de E/S permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador siendo el destino habitual de éstas líneas dispositivos simples como relés, LED etc.

#### **2.4.1.1.4. Reloj Principal**

Todo microcontrolador posee internamente de un circuito oscilador encargado de generar una onda cuadrada de alta frecuencia, la cual configura los impulsos de reloj usados en la sincronización de todas las tareas del sistema, éste circuito de reloj está incorporado en el microcontrolador.

El aumento o disminución de la frecuencia de éste reloj hace variar el tiempo en el cual se ejecutan las instrucciones, mientras más alta sea la frecuencias mayor rapidez en la ejecución sin olvidarse que esto conlleva a un mayor consumo de energía.

#### **2.4.1.1.5. Recursos Adicionales**

Los principales recursos adicionales incorporados en los Microcontroladores son:

- **Temporizadores o "Timers".**

Son empleados para controlar periodos de tiempo (temporizadores) y llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

- **Perro guardián o "Watchdog".**

El Perro guardián no es más que un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema, éste temporizador debe ser controlado por un programa el cual refresque e inicialice el Watchdog antes de que provoque el reset.

- **Conversor A/D Análogo/Digital (CAD).**

El conversor A/D procesa señales analógicas en las aplicaciones, disponen de un multiplexor que permite aplicar a la entrada del CAD varias señales analógicas directamente desde el circuito integrado.

- **Conversor D/A Digital/Análogo.**

Este conversor transforma los datos digitales que se obtiene del procesamiento del computador en su correspondiente señal analógica la cual sale por una de las líneas de E/S del circuito integrado.

- **Puertas de E/S digitales.**

Todo microcontrolador tiene líneas dedicadas a ser E/S digitales, generalmente estas líneas se agrupan de ocho en ocho formando Puertos y estos son configurados como Entrada o como Salida, cargando un 1 ó un 0 en el bit correspondiente.

- **Puertas de comunicación**

Ayudan a comunicarse con otros dispositivos externos, mediante distintas normas y protocolos. Entre los más destacados están:

- **UART** (Universal Asynchronous Receiver Transmitter), adaptador de comunicación serie asíncrona<sup>16</sup>.

---

<sup>16</sup> Asíncrona Sistema que no responde a un reloj constante.



- **USART** (Universal Synchronous Asynchronous Receiver Transmitter), adaptador de comunicación serie síncrona<sup>17</sup> y asíncrona.
- **USB** (Universal Serial Bus), que es un moderno bus serie para los Computadores.
- **CAN** (Controller Area Network)

#### **2.4.1.2. Arquitectura de Microcontroladores**

Inicialmente los Microcontroladores poseían la arquitectura Von Neumann, que con el tiempo ha sido un poco desplazada por la arquitectura Harvard, a continuación, se describen las dos:

##### **2.4.1.2.1. Arquitectura Von Neumann**

En esta arquitectura se define una sola memoria de datos e instrucciones conectada al CPU mediante un bus único de datos, direcciones y control.

##### **2.4.1.2.2. Arquitectura Harvard**

Es una arquitectura moderna en la cual se encuentran 2 memorias conectadas al CPU, una de datos y la otra de instrucciones, ambas tienen su propio bus, haciendo posible realizar operaciones de acceso es decir, de lectura y escritura en un mismo tiempo.

---

<sup>17</sup> Síncrona Este tipo de transmisión tiene la función de sincronizar los relojes existentes tanto en el emisor como en el receptor, de tal forma que estos controlan la duración del envío o recepción de información.

## **2.4.2. SENSORES Y ACTUADORES**

### **Sensor**

Es un dispositivo capaz de detectar magnitudes físicas o químicas para transformarlas en variables eléctricas que se puedan cuantificar y manipular, es así que para éste proyecto se utilizará los siguientes sensores: sensor fotovoltaico, detector de humo, detector de gas, detector de movimiento, sensor magnético.

### **Actuador**

Los actuadores son elementos que provocan algún efecto sobre un proceso automatizado, son capaces de generar una fuerza a partir de líquidos, de energía eléctrica o gaseosa. El actuador recibe la orden de un sensor o detector y da una salida para activar a un elemento final. Para este proyecto se utiliza como actuadores a: focos, bocina, cortina eléctrica, cerradura eléctrica.

#### **2.4.2.1. Sensor magnético**

Este sensor no es más que un interruptor magnético, es decir, se activa cuando un imán se le aproxima a 15m/m. y se desactiva, cuando este imán se separa a más de 25m/m.

##### **2.4.2.1.1. Partes de un sensor magnético:**

Consta de 2 partes que son:

- **Soporte electrónico:**

El soporte electrónico, trabaja de igual forma que un relé porque incorpora en su interior, mecanismos que hacen abrir y cerrar el circuito.

- **Soporte magnético:**

El soporte magnético, tiene en el interior dos imanes polarizados de forma que su campo se dirija de forma adecuada. En el caso de éste sensor el campo es emitido lateralmente al soporte.

#### **2.4.2.1.2. Principio de Funcionamiento**

Su funcionamiento se basa en el efecto que produce un par de láminas dentro de un campo magnético. Existen dos tipos, en uno de ellos los contactos permanecen abiertos cuando el soporte electrónico no está próximo al campo magnético, si se aproxima un imán las láminas se unen cerrando los contactos.

El segundo tipo es contrario al primero, es decir, dentro del campo magnético generado por el soporte los contactos están normalmente abiertos y al separarlo del imán o del soporte electrónico se unen.

#### **2.4.2.2. Sensor Fotovoltaico**

Este sensor es capaz de detectar o medir el nivel de luz en un espacio físico determinado y producir una señal de salida dependiendo de la intensidad de la luz a la que está expuesto.

El funcionamiento básico de estos sensores se basa en convertir la luz en una señal eléctrica, se construye básicamente de dos formas:

##### **2.4.2.2.1. Fotorresistencias**

También llamado LDR (Light Dependent Resistors) está basado en el efecto que produce un haz luminoso sobre un material semiconductor sensible a la luz, fotosensible (es decir, cuando más luz exista menos resistencia habrá), los valores que arroja el sensor varían totalmente cuando está completamente iluminado o se encuentra a oscuras.

Los valores son medidos en ohmios, y puede llegar a medir de 50 ohmios a 1000 ohmios (1K) en iluminación total y de 50K (50,000 Ohms) a varios megaohmios cuando está a oscuras.

El tiempo de respuesta de estos sensores es de aproximadamente 0.1 segundos.

Este tipo de sensores son muy aplicados en circuitos donde se necesita detectar la presencia o no de luz natural.

#### **2.4.2.2.2. Células fotovoltaicas**

Este tipo de células genera corriente cuando se recibe luz y el efecto que produce se llama fotovoltaico, ese tipo de células son poco utilizadas como sensores.

#### **2.4.2.3. Detector de movimiento**

Los sensores de movimiento pueden utilizar dos tipos de tecnología, una es la tecnología de los rayos infrarrojos y la otra las ondas ultrasónicas que se usan para “mapear” o captar en tiempo real los movimientos que se generen en un espacio determinado, donde el sensor se encuentre instalado.

Este tipo de sensores son uno de los más utilizados y reconocido dentro de la seguridad electrónica, porque su tamaño y funcionalidad lo hacen ideal para hacer un control de seguridad efectivo.

##### **2.4.2.3.1. Microondas**

Este tipo de sensores de espacio de microondas utilizan ondas de radio dentro del lugar en el que son instalados emitiendo estas ondas alternadamente, su único inconveniente es que son demasiado sensibles, es decir, cualquier tipo de interrupción que tengan las ondas de radio se activa el sensor, esto en algunos casos resulta un problema porque una leve brisa o un pequeño animal puede activar la alarma.

Una de las ventajas de estos sensores, es que sus ondas de radio son muy potentes y pueden atravesar fácilmente hasta una pared u otros objetos sólidos, haciéndolos a los sensores fáciles de ocultar y sin tener una línea de vista despejada los sensores pueden cumplir tranquilamente su función de supervisado.

Otro inconveniente de estos sensores es el de su instalación, debido a que sus ondas de radio son altamente potentes deben ser colocados en un lugar donde únicamente supervise las áreas en las que se desee detectar presencia, caso contrario el sensor puede detectar una intrusión cuando no está ocurriendo en realidad.

#### **2.4.2.3.2. Detectores Infrarrojos Pasivos de Movimiento**

Los sensores infrarrojos pasivos de movimiento fueron diseñados con el objetivo de eliminar las deficiencias de los sensores de ondas de radio y evitar así el problema de las falsas alarmas.

El problema de los falsos movimientos es resuelto por estos sensores al contar con una sensibilidad establecida, y de igual forma que detectan el movimiento están supervisando las variaciones de temperatura que se emiten en el lugar.

Imagine el siguiente ejemplo que explicará el funcionamiento de este tipo de sensores: el sensor está instalado en un cuarto cerrado con una temperatura ambiente de 20 °C (temperatura supervisada). Ahora cuando un intruso entre al área supervisada por el sensor se generará un aumento de temperatura (los seres humanos tienen una temperatura corporal de 36 °C), es ahí cuando el sensor infrarrojo pasivo de movimiento detecta ese cambio brusco de temperatura y activa la alarma.

Algo muy importante que se ha incorporado a estos sensores es la capacidad de distinguir entre horizontal y vertical haciendo posible eliminar la posibilidad de que un animal domestico que vive dentro de la casa encienda la alarma.

La única desventaja de éstos sensores es el reducido alcance que tienen es así que cuando se requiere supervisar una área extensa es necesario una mayor cantidad, motivo por el cual se necesita una mayor inversión.

#### **2.4.2.4. Detector de humo**

Este es uno de los sensores de mayor uso que ayuda a la prevención de incendios, un sensor de humo no es más que un sistema sensible a la presencia de humo o partículas de combustión que se puedan encontrar en el aire.

Los sensores de humo actúan conjuntamente con un sistema de alarma, su función básica es de dar un aviso previo a un incendio que esté por ocurrir al detectar la presencia de humo.

Este sistema de detección está compuesto básicamente de un sensor y un actuador, el sensor es el encargado de detectar la presencia de humo mientras que el actuador activa algún tipo de sistema de alarma o emite un sonido.

Existen dos tipos de sensores de humo, por efecto fotoeléctrico y por ionización<sup>18</sup>:

##### **2.4.2.4.1. Sensor de humo por efecto fotoeléctrico**

También llamados "ópticos", este tipo de sensores tienen una celda fotoeléctrica donde la iluminación de un metal (que es ve afectada por la presencia de humo), genera una débil corriente que activa algún actuador.

---

<sup>18</sup> La ionización es el proceso químico o físico mediante el cual se producen iones. Los iones son átomos o moléculas cargadas eléctricamente debido al exceso o falta de electrones respecto a un átomo o molécula neutra.

Este tipo de sensor es más sensible a los incendios de desarrollo lento es decir, incendios en los cuales aun no existe la presencia de fuego.

#### **2.4.2.4.2. Sensor de humo por ionización**

Detecta las variaciones en la corriente de iones debida a la presencia del humo.

Al producirse un incendio se generan iones en el aire, pero es difícil que estas partículas se mantengan así hasta cuando se encuentren con el detector, por este motivo el aire se ioniza dentro a esto se le llama Cámara de Ionización y de esta forma el sensor sabe cuándo activar la alarma.

Esta clase de sensores tienen una mejor reacción cuando los incendios son de rápido desarrollo y existe la presencia de fuego, y tienen un grado mayor de sensibilidad ante el humo que emite el cigarrillo.

#### **2.4.2.5. Detector de gas**

Es un dispositivo que detecta la presencia de gas combustible o de humo en el aire, dependiendo de la concentración en la que se encuentre se emite una señal que puede ser óptica, acústica o un corte automático de gas, esto depende de la utilidad que se le quiera dar al sensor.

Su funcionamiento se basa en la resistencia interna del detector dependiendo de la concentración de gas en el lugar, soporta de 300 a 10.000 ppm<sup>19</sup> (partes por millón). La interfaz de estos dispositivos es relativamente sencilla porque su voltaje es analógico, además funciona a temperaturas de -20 a 50° C y su consumo de energías es menos de 150mA a 5V.

---

<sup>19</sup> Partes por millón (ppm), es una unidad de medida de concentración de una solución, ésta se refiere a la cantidad de mg (miligramos) que se encuentra en un kg de disolución.

#### **2.4.2.6. Relés**

Los relés son dispositivos electromecánicos que funcionan como un interruptor que se acciona mediante un electroimán.

Estos dispositivos se controlan mediante un circuito eléctrico en el que actúan una bobina y un electroimán, los cuales activan contactos que permiten abrir o cerrar otros circuitos independientes. Controlan circuitos que posean una mayor potencia de salida que los de entrada.

Los relés están compuestos por dos tipos de contactos, de trabajo (se cierran cuando la bobina se alimenta) y de reposo (se cierran cuando la bobina no se alimenta).

#### **2.4.3. PROTOTIPO CORTINAS ELÉCTRICAS**

Las cortinas eléctricas hoy en día son una solución adoptada por los hogares que ven en ellas una manera más práctica y cómoda de automatizar una tarea que se ha llevado de forma manual y rutinariamente en el hogar.

Éste prototipo es muy simple de montar e instalar, se realiza la apertura o cierre de las cortinas dependiendo de varios factores que se desee tomar en cuenta, estos son: presencia y ausencia de luz natural, privacidad del hogar, exceso de luz solar que ingresa, etc.

Todo el sistema que implica las cortinas eléctricas debe ser controlado mediante otro sistema inalámbrico que permita su accionamiento desde otro lugar del hogar, estos dos sistemas interactúan conjuntamente para hacer uno solo y realizar la acción para la cual fueron diseñados.



Su funcionamiento está basado en un circuito eléctrico en el cual actúa un motor que sirve para el mecanismo de cierre o apertura, montado en una estructura que le permite mover los rieles en las cuales se encuentran sujetas las cortinas. Este mecanismo eléctrico actúa mediante la acción de otro sistema inalámbrico que lo activa o desactiva.

#### **2.4.4. CÁMARA IP**

La cámara IP o también llamada Web o de red, son cámaras diseñadas con el fin de enviar video y en algunos casos audio a través de una red IP local mediante un Switch, Router o ya sea por Internet.

Su mayor utilidad es ver las imágenes que capta y envía la cámara a través de Internet, esta cámara puede instalarse en cualquier parte que posea una conexión a internet, su direccionamiento ip es fijo o dinámico.

El acceso a las imágenes de la cámara IP pueden o no ser restringidas, según sea la necesidad para las que fueron montadas.

Son utilizadas principalmente para objetivos de seguridad y vigilancia de hogares, locales comerciales, empresas, etc., interactúan con otros dispositivos como son los sensores de movimiento.

##### **2.4.4.1. Componentes de la Cámara IP**

Una cámara IP está compuesta de:

- Una cámara de video tradicional con sus típicos componentes lentes, sensores, procesador digital de imagen.
- Un sistema de compresión de imagen, para comprimir las imágenes enviadas a formatos adecuados como es MPEG4.
- Un sistema de procesamiento (CPU, FLASH, DRAM<sup>20</sup> y un módulo Wireless<sup>21</sup> ETHERNET/WIFI<sup>22</sup>). Lo que hace este sistema es

---

<sup>20</sup> Dynamic random access memory – (Memoria de acceso aleatorio dinámica).

encargarse de la gestión de las imágenes, la gestión de envío de alarmas, autorización para visualizar las imágenes.

Todo esto convierte a la cámara IP en un equipo autónomo que tan solo al conectarlo a un Router ADSL<sup>23</sup> y suministrarle energía eléctrica está en la capacidad de enviar imágenes de desplazamiento a cualquier parte donde se disponga de Internet.

#### **2.4.5. PANTALLA TÁCTIL**

Se controlan básicamente con Microcontroladores dependiendo de la utilidad que se le quiera dar al GLCD, es una lamina que se coloca encima del GLCD para lograr obtener puntos de presión (coordenadas).

##### **2.4.5.1. Funcionamiento de una Pantalla Táctil**

La pantalla táctil realiza su función de distintas maneras ya sea por, reconocimiento óptico, localización de pulsos de ondas, combinaciones eléctricas con cristal o infrarrojo.

Cualquiera de estas maneras debe ser manipulada por el dedo o un objeto, es así, que cuando se tiene contacto con ella, es enviado un pulso hacia los sensores encargados de detectar las coordenadas exactas, se las envía al controlador, quien a su vez interpreta los pulsos recibidos y los convierte en pulsos digitales.

---

<sup>21</sup> Se denomina Wireless a las comunicaciones inalámbricas, en las que se utilizan modulación de ondas electromagnéticas, radiaciones o medios ópticos.

<sup>22</sup> Abreviatura de Wireless Fidelity

<sup>23</sup> ADSL son las siglas de Asymmetric Digital Subscriber Line ("Línea de Abonado Digital Asimétrica").

## **2.5. SOFTWARE DE DESARROLLO**

### **2.5.1. LENGUAJE DE PROGRAMACIÓN**

Los lenguajes de programación son idiomas artificiales diseñados para controlar el comportamiento de una máquina tanto físico y lógico, de ésta manera se expresa algoritmos, cálculos, modos de comunicación humana con suma precisión.

Están compuestos por un sin número de reglas, símbolos, sintaxis, elementos, expresiones definidas previamente que forman la base lógica y elemental del lenguaje.

Análogamente un lenguaje de programación se acerca a la forma humana de comunicarse por tal motivo se conocen como lenguajes de alto nivel, al utilizar palabras, sentencias y estructuras semejantes al lenguaje natural.

El programador es el encargado de hacer que el lenguaje de programación se convierta en un programa listo para usar según las necesidades o la solución que se plantea.

Un lenguaje de programación necesita convertirse en un programa informático que controle o ejecute acciones, los siguientes pasos muestran cómo se consigue esto:

- Un desarrollo lógico del programa.
- Codificación del programa es decir, la escritura de la lógica del programa por medio del lenguaje de programación.
- Compilación del programa para convertirlo en lenguaje de máquina.
- Depuración del programa.

### **2.5.1.1. Tipos de Lenguajes de Programación**

Los lenguajes de programación se dividen en dos categorías:

#### **2.5.1.1.1. Lenguajes interpretados**

Los lenguajes de programación al poseer un lenguaje diferente al comprendido por la máquina requieren de un intérprete capaz de traducir las instrucciones para que sean entendidas por el procesador. De esta manera requieren de un programa auxiliar que ayude a realizar ésta tarea y a esto se conoce como “el intérprete”.

#### **2.5.1.1.2. Lenguajes compilados**

En éste tipo de lenguajes existe un compilador que se encarga de crear un nuevo archivo independiente que no necesita de otro para ejecutarse, a esto se llama archivo ejecutable, volviendo la ejecución del código compilado más rápido.

La desventaja que presenta éste tipo de lenguajes es que cada modificación que se realiza al código fuente, requiere la compilación para así ver los cambios, y su ventaja fundamental es la seguridad del código fuente, esto no ocurre con el lenguaje interpretado.

#### **2.5.1.1.3. Lenguajes intermediarios**

Aquí entran los lenguajes que poseen las características de los dos tipos de lenguajes mencionados, debido a que en alguna parte tiene una fase de compilación en un archivo escrito en un lenguaje no conocido o que difiere del código fuente y no es ejecutado, entonces requeriría necesariamente de un intérprete.

### **2.5.2. BASE DE DATOS**

Una base de datos es un conjunto de información almacenada y ordenada en forma de registros (filas) y campos (columnas), mediante las cuales se recupera cualquier tipo de información como por ejemplo: referencias, documentos textuales, imágenes, datos estadísticos, datos que arroje un sistema etc.

Las bases de datos se han convertido prácticamente en el soporte esencial para cualquier sitio Web, intranet y otros sistemas de información en los que se requiera manejar grandes, medianos o pequeños volúmenes de información, a la cual pueden acudir cualquier tipo de usuario para realizar consultas o acceder a información que sea de su interés, convirtiendo en estos tiempos a la base de datos en una herramienta imprescindible en cualquier sistema informático.

Características básicas que debe poseer una bases de datos:

- Redundancia mínima de información.
- Acceso concurrente por varios usuarios.
- Integridad de la información.
- Consultas complejas optimizadas.
- Seguridad de acceso a la información.
- Auditoria del acceso a la información.
- Respaldo y recuperación de la información.
- Acceso a la base de datos a través de herramientas de programación.

### 2.5.3. SERVIDORES

#### 2.5.3.1. Servidor Web

Un servidor web es un programa que se ejecuta en un computador en espera de peticiones de ejecución realizadas por otros usuarios a través de Internet, el servidor web se comunica con el usuario mediante el protocolo HTTP y contesta a las peticiones enviando como respuesta código HTML de una página web o información de todo tipo que será mostrada en el navegador.

Cualquier computador puede servir como servidor web siempre y cuando se encuentre conectado a internet, claro que tener un servidor web local implica que uno mismo debe gestionar todos los niveles técnicos, tanto de acceso como de conexión a la red, y comprobar que todo funciona correctamente. Por otro lado al contratar un servidor web con alguna compañía se opta por dos opciones: un “servidor dedicado” el cual es único y exclusivo para el sitio web del cliente que lo contrata, o un “servidor compartido” es decir, que este servidor será usado por varios clientes compartiendo sus recursos.

En conclusión, se define el funcionamiento de un servidor web de la siguiente manera:

- El usuario digita en su navegador la URL<sup>24</sup> `http://www.servidor.com/archivo.html` donde “http” es el protocolo utilizado para hablar con el servidor `www.servidor.com`, y “archivo.html” es el fichero al que se quiere acceder.
- El navegador se comunica con el servidor para traducir el nombre de la URL `www.servidor.com` para convertirlo en una dirección IP, la cual es usada para conectar al servidor Web.
- El navegador entonces forma una conexión con el servidor en esa dirección IP, que normalmente se genera en el puerto 80.

---

<sup>24</sup> Uniform Resource Locator - Localizador Uniforme de Recursos.

- Siguiendo con el protocolo http, el navegador envía una petición al servidor preguntando por el fichero archivo.html.
- El servidor envía el código HTML de la página Web solicitada al navegador.
- El navegador lee e interpreta el código HTML que recibió y lo muestra en su navegador.

#### **2.5.3.2. Servidor de correo Web (Webmail)**

Un Webmail es un entorno de correo electrónico independiente del computador, basado en la WWW (World Wide Web) dentro del cual se gestiona el correo electrónico de forma rápida y simple, mediante cualquier computador que tenga conexión a internet sin tener que realizar ninguna configuración previa.

Solo basta con hacer un registro previo y rápido de una cuenta de correo que tendrá un nombre de usuario y contraseña.

El Web mail se convierte en una gran alternativa para acceder a un correo electrónico en un computador que se encuentre en lugares públicos como cibercafés, hoteles, bibliotecas donde no es posible realizar configuraciones adicionales o utilizar programas distintos sino solo a través de un navegador de Internet.

Este tipo de servidores proporcionan seguridad en el envío y recepción de correos, puesto que siempre viajan de forma cifrada a través de la red.

Además de enviar y recibir mensajes, los Webmail tienen otras características como son: espacio para el almacenamiento de correo (depende del Webmail utilizado), autor respuestas, mensaje de vacaciones, filtros, propagandas.

Entre los Webmail más conocido y usados se mencionarán a: Hotmail, yahoo, GMAIL.

## **CAPÍTULO 3**

### **ANÁLISIS DE REQUERIMIENTOS**

En este capítulo, se realiza un análisis previo al diseño de la solución, se dará a conocer cuáles son los requerimientos adecuados y mediante el estudio de éstos se escogerá cual es el más apropiado y el que se ajusta más a este proyecto de tesis.

#### **3.1. ANÁLISIS DE TECNOLOGÍAS INALÁMBRICAS**

##### **La Radio Frecuencia**

Es un término que hace referencia a la corriente alterna (AC), que al ser alimentada a una antena genera un campo electromagnético que sirve para la transmisión de información de manera inalámbrica.

Este tipo de frecuencias cubren un rango del espectro electromagnético que van desde los 9 KHz hasta los miles de Gigahertz (GHz).

Según se vaya incrementando la frecuencia de las ondas electromagnéticas por fuera del espectro de RF, su energía toma la forma de ondas infrarrojas (IR), visibles, ultravioletas (UV), rayos X y rayos gama.

Muchos dispositivos inalámbricos emplean los campos de la RF, tales como: teléfonos, celulares, radios, televisiones, intercomunicadores o servicios de radio con comunicación bi-direccional, etc.

Mediante las tecnologías de radio frecuencia se crean redes inalámbricas para transmitir información que se requiera y de esta forma eliminar el exceso de cables, convirtiendo a la RF en una manera segura, rentable, accesible de transmitir información.



### **3.1.1. BLUETOOTH**

Bluetooth tecnología de comunicaciones inalámbricas definida bajo el estándar IEEE 802.15.1 utilizada para la transmisión de voz y datos entre diferentes dispositivos que admiten este estándar mediante radiofrecuencia en la banda ISM de los 2,4 a 2,48 GHz con la posibilidad de transmitir en Full Duplex a una distancia de 10 metros (incluso atravesando objetos o paredes) de forma segura y sin licencia de corto rango.

Esta tecnología es muy cotizada y prácticamente hoy en día se ha convertido en un estándar obligado en dispositivos de telecomunicaciones personal como son: PDA, teléfonos móviles, computadoras portátiles, impresoras o cámaras digitales.

Entre las principales características con las que cuenta Bluetooth se citarán las siguientes:

- Facilita la comunicación entre equipos móviles y fijos.
- Eliminar cables.
- Creación de pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.
- Trabaja con un sin número de aplicaciones bajo este estándar como aplicaciones de control remoto.
- En consolas de juegos como Sony PlayStation 3 y Nintendo Wii incorporan Bluetooth, que les permite utilizar mandos inalámbricos.

### **3.1.2. INFRARROJO**

La comunicación por infrarrojos es utilizada cotidianamente sin darse cuenta, se utiliza universalmente en los mandos a distancia de la televisión, equipos de sonido o equipos de video, presentando una alternativa rentable para la comunicación punto a punto entre dispositivos electrónicos.

En la actualidad la mayoría de teléfonos celulares están equipados con ésta tecnología de comunicación que les permite conectarse a redes de acceso telefónico.

La comunicación mediante infrarrojo se encuentra limitada por el espacio y los obstáculos al tener una longitud de onda demasiado pequeña, esto hace que no se pueda propagar de la misma manera como hacen las tecnologías que utilizan señales de radio, por este motivo que este tipo de comunicación se implementa en redes en espacios reducido como solo cuarto o a lo mucho entre un piso.

Toda ésta comunicación se realiza a través de un “transreceptor<sup>25</sup>” (dispositivo capaz de emitir y detectar luz infrarroja) utilizado para enviar datos codificados mediante un protocolo, y de igual manera al otro lado un transreceptor para recibir estos datos.

#### **3.1.2.1. Tipos de comunicación infrarroja**

Existen 3 tipos de modos de comunicación por infrarrojos y estos son:

Punto a Punto: Tanto el emisor como el receptor deben estar lo más cercanos posibles y en una alineación correcta, es decir, que requieren una línea de vista para comunicarse.

##### **3.1.2.1.1. Cuasi difuso**

Los dispositivos se comunican por intermedio de superficies reflejantes, solo es necesaria la línea de vista con estas superficies.

##### **3.1.2.1.2. Difuso**

La señal infrarroja de salida debe llenar completamente toda la habitación o lugar donde se va a transmitir, haciendo innecesaria la línea de vista, permitiendo que los demás dispositivos tengan libertad de movimiento.

---

<sup>25</sup> Dispositivo capaz de emitir y detectar luz infrarroja.

### 3.1.3. ZIGBEE

Es un estándar definido por la IEEE 802.15.4 cuya ventajas superan enormemente a tecnologías como Bluetooth, trabaja en la banda libre 2.4 GHz, posee seguridad en sus módulos y en la transmisión de información, el alcance de transmisión mediante Zigbee varía entre los 100 mts a los 1500 mts.

Esta tecnología permite crear redes con una gran cantidad de nodos distribuidos en distintos lugares y con varias topologías: punto a punto, árbol, malla.

Zigbee está siendo utilizada en su mayoría en dispositivos electrónicos domésticos e industriales para su automatización, debido a que representa una tecnología de largo alcance con un bajo consumo de energía y largo tiempo de vida útil de sus baterías.

En la siguiente tabla, se hace una comparación de las tecnologías inalámbricas descritas previamente para presentar sus ventajas, desventajas y así posteriormente escoger cuál de ellas se utilizará.

TECNOLOGÍA	VENTAJAS	DESVENTAJAS
BLUETOOTH	<ul style="list-style-type: none"><li>▪ No requiere línea de vista</li><li>▪ Los dispositivos pueden estar en movimiento al momento de comunicarse</li><li>▪ Permite la generación de redes</li><li>▪ Realizar el registro y descubrimiento de los servicios disponibles en la red</li><li>▪ Mayor alcance de transmisión a diferencia de otras tecnologías</li></ul>	<ul style="list-style-type: none"><li>▪ Mayor uso de batería en modo visible</li><li>▪ Navegación por internet lenta</li><li>▪ Seguridad</li></ul>
INFRARROJO	<ul style="list-style-type: none"><li>▪ Utilizada en la mayoría de dispositivos electrónicos</li><li>▪ Alta seguridad</li></ul>	<ul style="list-style-type: none"><li>▪ Requiere línea de vista</li><li>▪ Las frecuencias de la banda del infrarrojo no permiten la penetración a través de paredes u obstáculos</li><li>▪ Alcance de transmisión</li></ul>
ZIGBEE	<ul style="list-style-type: none"><li>▪ No requiere línea de vista</li><li>▪ Largo alcance especialmente en sus módulos Xbee PRO</li></ul>	<ul style="list-style-type: none"><li>▪ Tasa de transferencia es baja</li><li>▪ Solo manipula información corta</li><li>▪ No es compatible con Bluetooth debidos a su baja tasa de</li></ul>

	1500mts <ul style="list-style-type: none"> <li>▪ Bajo consumo de Energía</li> <li>▪ Larga vida útil</li> <li>▪ Seguridad</li> <li>▪ Varias Topologías</li> <li>▪ Un gran número de nodos dentro de sus redes 65.000 nodos</li> <li>▪ Escalabilidad de red</li> <li>▪ Agilidad de Frecuencia</li> <li>▪ 128 bits AES de cifrado</li> <li>▪ Económicos</li> </ul>	transferencia y a su capacidad de soporte para nodos.
--	---	---

**Tabla 1: Ventajas y desventajas Tecnologías Inalámbricas**

Analizando la tabla anterior se concluye que Bluetooth y Zigbee serán las tecnologías que se usarán en el proyecto de tesis, la primera para manejar una aplicación móvil en dispositivos tales como celulares, que posean entre sus características la tecnología Bluetooth.

Por otro lado, Zigbee será la tecnología que se implementará en los módulos de control y será utilizada para transmitir información entre ellos de forma que puedan ser gestionarlos y monitorearlos.

## **3.2. ANÁLISIS DE SOFTWARE**

Se analizará los diferentes programas con el cual será implementado el sistema.

### **3.2.1. GESTOR DE BASES DE DATOS**

Se analizarán 4 gestores de Base de Datos y son los siguientes:

### **3.2.1.1. SQL Server 2008**

Microsoft SQL Server 2008 constituye la mejor alternativa presentada por Microsoft como gestor de base de bases de datos basado en el modelo relacional. Utiliza los lenguajes T-SQL<sup>26</sup> y ANSI SQL<sup>27</sup> para consultas.

SQL Server 2008 ofrece una plataforma de datos completa, segura, administrable y escalable, donde los desarrolladores crean aplicaciones nuevas, capaces de almacenar y consumir cualquier tipo de datos en cualquier dispositivo.

Es una herramienta diseñada para ser utilizada únicamente dentro de la plataforma Windows, existen varias versiones de SQL Server 2008 y dependiendo de la versión que se desee usar varía sus características y precio de licencia.

#### **El precio de la licencia depende de:**

- Número de procesadores a usar en el servidor.
- Tipo de versión.
- Número de clientes que van a acceder al servidor, esto se conoce como licencias de Acceso.
- Plataforma 32 bit o 64 bit.

Es así que SQL Server 2008 se encuentra en versiones Enterprise o Standard, con un costo de sus licencias que varía entre los 2300 y 6000 dólares, de igual manera existe la versión Developer que es una buena alternativa para empresas de desarrollo de software, el precio de su licencia es de 50 dólares.

---

<sup>26</sup> Transact-SQL es una extensión del lenguaje SQL, propiedad de Microsoft y Sybase

<sup>27</sup> Instituto Nacional Estadounidense de Estándares SQL

### 3.2.1.2. Oracle

Oracle es un sistema de gestión de base de datos relacional desarrollada por Oracle Corporation es una herramienta cliente/servidor para la gestión de base de datos, soporta varias plataformas como Windows, Unix, y las últimas versiones Oracle han sido certificadas para trabajar bajo GPL<sup>28</sup> GNU<sup>29</sup>/Linux.

Su dominio en el mercado de servidores empresariales es casi total, es un producto vendido a nivel mundial, debido a su gran potencia y su elevado costo únicamente es adquirido por empresas grandes y multinacionales.

Oracle es uno de los gestores de bases de datos más completos que existe porque cuenta con:

- Soporte de transacciones.
- Alto grado de estabilidad.
- Escalabilidad.

Se observa que Oracle ofrece muchas ventajas, pero de igual forma presenta desventajas como son: la necesidad de ajustes, es fácil caer en el error de pensar que solo basta con instalar Oracle en el servidor y conectarlo con las aplicaciones clientes, pues no es así porque un Oracle mal configurado se vuelve demasiado lento.

El mayor inconveniente que se observa en Oracle son sus elevados precios no solo en sus licencias sino el costo de la información que se requiere tener para la instalación y administración. Como se señalaba antes éste no es inconveniente para empresas multinacionales que lo adquieren porque cuentan con los recursos suficientes para trabajar con Oracle y sin duda les ofrece mucho más ventajas que otros generadores de Bases de Datos.

---

<sup>28</sup> Licencia Pública General es una licencia libre, sin derechos para software y otro tipo de trabajos.

<sup>29</sup> En inglés "gnu" (en español "ñu"), es un acrónimo recursivo de «GNU No es Unix».

### 3.2.1.3. MySQL

MySQL es sin duda uno de los sistemas de administración de bases de datos relacionales Open Source mas popular que existe y debido a su constante crecimiento se está convirtiendo en un competidor más directo de los grandes gestores de bases de datos como Oracle.

MySQL fue desarrollado MySQL AB una empresa sueca, que mantiene el Copyright<sup>30</sup> del código fuente del servidor SQL, y también de la marca, pese a ser software libre MySQL AB distribuye una versión comercial que únicamente se diferencia a la versión libre de MySQL en el soporte técnico que se da.

Este gestor fue escrito en C y C++, se adapta a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java, por éste motivo es una herramienta de gran aceptación por su rapidez, fácil uso, portabilidad entre sistemas, fácil instalación y configuración, ágil gestión de usuarios y contraseñas, manteniendo un alto nivel de seguridad en los datos.

Una de las mejores ventajas que ofrece MySQL y que ha llevado a este gestor a un desarrollo positivo y rápido es su condición de Open Source que además de hacer gratuito su uso, se tiene la libertad de modificar su código fuente, por ende sus continuas actualizaciones la convierten en la herramienta más usada por programadores y desarrolladores orientados a Internet.

---

<sup>30</sup> El copyright es una forma de protección legal que otorga a las personas que crean y producen contenido derechos exclusivos para controlar determinados usos de ese contenido. Entre los tipos de contenido protegido por copyright se pueden incluir, por ejemplo, canciones, obras de arte, vídeos, videojuegos, libros y películas.

Una de las pocas desventajas que se encontró en MySQL es que a diferencia de otras herramientas, no soporta el uso de procedimientos almacenados y triggers, no es intuitivo como otros gestores y la mayoría de utilidades que ofrece MySQL no está documentada.

#### **3.2.1.4. PostgreSQL**

Sistema de gestión de base de datos relacional con licencia libre por tal motivo no es manejado por una sola empresa sino por un grupo de desarrolladores que trabajan de forma desinteresada generando nuevas actualizaciones y mejoras a este gestor.

PostgreSQL es considerado como la base de datos de código abierto más avanzada del mundo, al proporcionar características muy avanzadas que solo se encontraría en gestores propietarios como es Oracle.

Características de PostgreSQL:

- Capaz de manejar rutinas y reglas complejas.
- Soporte multiusuario.
- Optimización de consultas, herencia y arrays.
- Soporta tipos de datos soportados por el usuario.
- Integridad referencial que garantiza la validez de sus datos.
- Disparadores.
- Procedimientos Almacenados.

Existen pocas desventajas por parte de este gestor de base de datos entre ellas se mencionará: que en relación a MySQL es mucho más lento al momento de realizar actualizaciones e inserciones debido a que posee cabeceras de transacción.



En la siguiente tabla, se muestra las ventajas y desventajas de los gestores de bases de datos analizados anteriormente.

HERRAMIENTA	VENTAJAS	DESVENTAJAS
SQL Server	<ul style="list-style-type: none"> <li>▪ Escalable</li> <li>▪ Administrable</li> <li>▪ Segura</li> </ul>	<ul style="list-style-type: none"> <li>▪ Precio de las licencias</li> <li>▪ No es multiplataforma</li> </ul>
Oracle	<ul style="list-style-type: none"> <li>▪ Multiplataforma</li> <li>▪ Es potente</li> <li>▪ Altamente Estable</li> <li>▪ Soporte de transacciones</li> </ul>	<ul style="list-style-type: none"> <li>▪ Alto precio en las licencias</li> <li>▪ Difícil configuración</li> </ul>
MySQL	<ul style="list-style-type: none"> <li>▪ Open Source</li> <li>▪ Portabilidad entres sistemas operativos</li> <li>▪ Fácil instalación y configuración</li> <li>▪ Gestión de Usuarios</li> <li>▪ Continuas Actualizaciones</li> </ul>	<ul style="list-style-type: none"> <li>▪ No soporta procedimientos almacenados</li> <li>▪ No es intuitivo</li> <li>▪ Utilidades no documentadas</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>▪ Multiplataforma</li> <li>▪ Disparadores</li> <li>▪ Procedimientos Almacenados</li> <li>▪ Licencia libre</li> </ul>	<ul style="list-style-type: none"> <li>▪ Al realizar sentencias Insert y Update es un poco más lento a relación de otros.</li> <li>▪ Información sobre sus fortalezas</li> <li>▪ Todavía no es muy difundido</li> </ul>

Tabla 2: Ventajas y desventajas Servidores de Bases de Datos

Después de éste rápido análisis entre estos grandes gestores de base de datos y contemplando sus ventajas – desventajas y sobre todo enfocado los requerimientos, alcances, costo-beneficio que éste proyecto de tesis persigue se ha optado por utilizar la herramienta MySQL en su versión 5.1.37, manejada a través de la herramienta phpMyAdmin 3.2.0.1 del servidor XAMPP<sup>31</sup>, la cual se acopla a las necesidades del proyecto y es la mejor opción para el desarrollo de una base de datos relacional, que de las garantías técnicas que el sistema requiere soportando la información que en ella se almacenara de forma ordenada y segura.

<sup>31</sup> XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (cualquier sistema operativo), **A**pache, **M**ySQL, **P**HP, **P**erl.

A pesar de las desventajas que se tiene con MySQL estas se han podido conllevar para que no sean un obstáculo en el manejo de esta herramienta y sobrellevar estos problemas de manera que no afecte ni disminuya el rendimiento del sistema.

### **3.2.2. LENGUAJES DE PROGRAMACIÓN APLICACIÓN WEB**

#### **3.2.2.1. Lenguaje PHP**

PHP es un lenguaje de programación de alto nivel multiplataforma embebido en páginas HTML que se ejecutan en el lado del servidor. Este lenguaje de programación cuenta con una gran cantidad de librerías y mucha documentación sobre ellas.

PHP o Hypertext Pre-processor es un lenguaje que se ejecuta en el servidor web, momentos antes de ser enviada la página web al cliente mediante Internet, es decir, todos los procesos como acceso a base de datos, conexiones de red etc., son ejecutadas en el propio servidor y únicamente el cliente recibe una página con el código HTML resultado de la ejecución del PHP, haciendo una programación segura y muy confiable.

Al contar PHP con una extensa gama de librerías lo hace un lenguaje de programación capaz de realizar aplicaciones web dinámicas de alto nivel y muy complejas, las cuales son ejecutadas y procesadas en pocos segundos debido a la rapidez de éste lenguaje de programación.

La versión más reciente de este lenguaje es la 5.3.3, del 22 de julio de 2010, versión en la cual se incluyeron mejoras para la seguridad y estabilidad de las páginas web.

La gran ventaja que se destaca de PHP es que tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, como son:

UNIX, Linux, Mac OS X y Windows, de esta forma interactuar con los servidores web más utilizados como Apache, e ISAPI<sup>32</sup>.

### **3.2.2.2. Lenguaje Java**

Java es un lenguaje de programación, desarrollado por Sun Microsystems, es un lenguaje orientado a la programación de objetos, es un lenguaje multiplataforma es decir, puede ser creado en una plataforma y ejecutado en otra muy diferente sin la necesidad de modificar nada.

Hoy en día se ha convertido en uno de los lenguajes de programación líderes en el mercado, con las ventajas que ofrece lo hacen más apreciado. Dentro de java se desarrolla tanto aplicaciones simples como son programas de consola, aplicaciones complejas, aplicaciones web, que son ejecutadas desde un servidor.

Java está siendo utilizado en la actualidad para crear en él aplicaciones relativamente modernas como son foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.

Este lenguaje está presente no solo en ordenadores sino en dispositivos móviles, agendas y en general para cualquier otro dispositivo digital, presentando aplicaciones muy potentes y eficientes.

Java al ser un lenguaje de programación con licencia libre lo ha llevado a desarrollarse enormemente gracias a los aportes de la comunidad de desarrolladores que lo utilizan y van haciendo cada vez más un lenguaje estable, seguro y confiable, es así que Java se utiliza en todos lados, no solo en la web con páginas web muy sofisticadas, sino en

---

<sup>32</sup> Internet Service API es una interfaz de programación de aplicaciones (API) para el servidor web de Microsoft, IIS (Internet Information Server).

distintos equipos informáticos como dispositivos móviles, agendas, PDA<sup>33</sup> etc.

Java fue desarrollado principalmente para cumplir con las siguientes exigencias:

- Debe usar una metodología de programación orientada a objetos.
- Permitir la ejecución de un mismo programa en múltiples sistemas operativos (multiplataforma).
- Incluir soporte para trabajo en red.
- La capacidad de ejecución de su código en sistemas remotos de forma segura.
- Fácil de usar tomando lo mejor de otros lenguajes de programación orientados a objetos, como C++.
- Manejo de información y conexión segura con servidores de base de datos de una forma sencilla y rápida.

Para entender un poco más de cómo Java trabaja, se debe decir que éste lenguaje de programación consta de:

- El lenguaje de programación, mismo.
- La máquina virtual de Java o JRE el cual hace de java un lenguaje portable.
- El API Java.

## **JRE**

JRE acrónimo de Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java es el software indispensable para la ejecución de cualquier aplicación desarrollada en Java.

La función del JRE es servir de intermediario entre una aplicación creada en Java y el sistema operativo que se esté usando el usuario,

---

<sup>33</sup> PDA, del inglés personal digital assistant (asistente digital personal), es un computador de mano originalmente diseñado como agenda electrónica.

haciendo posible que cualquier sistema operativo que cuente con el JRE pueda ejecutar aplicaciones Java de la misma forma.

El JRE también se encuentra en el SDK<sup>34</sup> o JDK (Java Development Kit) que ofrece Sun Microsystem, acompañado de varias herramientas como el compilador de Java, Javadoc<sup>35</sup> para generar documentación, Java API<sup>36</sup>, visor de applets<sup>37</sup>.

## APIs

Sun define tres plataformas para cubrir los distintos entornos de aplicación Java. De esta manera ha distribuido muchos API (Application Program Interface) de forma que pertenezcan a cada una de las plataformas siguientes:

- **Java ME** (Java Platform, Micro Edition) o J2ME orientada a entornos de limitados recursos, como teléfonos móviles, PDAs etc.
- **Java SE** (Java Platform, Standard Edition) o J2SE para estaciones de trabajo.
- **Java EE** (Java Platform, Enterprise Edition) o J2EE orientada a entornos distribuidos empresariales o de Internet.

---

<sup>34</sup> Software Development Kit - Kit de desarrollo de software. Un SDK es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo.

<sup>35</sup> Genera documentación de código fuente en formato HTML.

<sup>36</sup> Java API (Application Program Interface) es un conjunto de librerías que permiten el desarrollo de aplicaciones en Java, brinda funciones como por ejemplo: Creación y manejo de elementos de GUI, manejo de archivos, funciones de red, comunicación entre programas.

<sup>37</sup> Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.

### **3.2.2.3. Lenguaje Visual Basic.Net 2008**

Es un lenguaje de programación orientado a objetos considerado la evolución del Visual Basic 6.0 pero ahora implementada sobre el Framework .NET. En esta versión .Net Visual Basic sufrió grandes cambios implementando modernas y avanzadas herramientas que hacen de éste un lenguaje de programación moderno, seguro y confiable.

Visual Basic .Net al igual que todos los lenguajes de programación basados en .NET, requieren del Framework .NET para ejecutarse dentro de Windows.

No es un lenguaje multiplataforma cuando se trabaja bajo Visual Studio, pero a pesar de esto es un lenguaje muy estable, de fácil manejo y aprendizaje, su sintaxis es básicamente la misma que se utilizaba en Visual Basic 6.0, claro con las nuevas ventajas que trae .Net como el control estructurado de excepciones, la programación orientada a objetos, la metodología de programación y nuevas bibliotecas.

Una de las ventajas que fue implementada a esta versión de VB .Net es que se especifica el Framework en el cual se desea compilar las aplicaciones creadas es así que se escoge entre:

- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5

Para el desarrollo de aplicaciones web cuenta con ASP.Net (Active Server Pages) un poderoso Framework para aplicaciones web desarrollado y comercializado por Microsoft, usado para construir sitios web dinámicos, aplicaciones web y servicios web XML<sup>38</sup>.

---

<sup>38</sup> XML son las siglas de Extensible Markup Language (lenguaje de marcas extensible).

Las aplicaciones creadas en ASP.Net desarrolladas en Visual Basic.Net pueden ser publicadas mediante el IIS (Internet Information Services) que ofrece una serie de servicios como FTP , SMTP , NNTP y HTTP/HTTPS , para computadoras que trabajan bajo Windows, es así que al trabajar como servidor web tiene la capacidad de publicar estas páginas web tanto local como remotamente.

En la siguiente tabla, se analiza las ventajas y desventajas de los Lenguajes de Programación para su posterior comparación y selección.

LENGUAJE	VENTAJAS	DESVENTAJAS
PHP	<ul style="list-style-type: none"> <li>▪ Open Source</li> <li>▪ Programación Orientada a Objetos</li> <li>▪ Manejo de Excepciones</li> <li>▪ Propia y extensa biblioteca de funciones</li> </ul>	<ul style="list-style-type: none"> <li>▪ Creación de código desordenado y difícil mantenimiento</li> <li>▪ Difícil de optimizar</li> </ul>
JAVA	<ul style="list-style-type: none"> <li>▪ Multiplataforma</li> <li>▪ Lenguaje más utilizado</li> <li>▪ Programación Orientada a Objetos</li> <li>▪ Confiable, seguro, estable</li> <li>▪ Open Source</li> <li>▪ Herramientas de desarrollo libre</li> <li>▪ Trabaja con varios servidores web</li> <li>▪ Múltiples librerías</li> <li>▪ Fácil aprendizaje</li> </ul>	<ul style="list-style-type: none"> <li>▪ Velocidad: al tener que ser ejecutado mediante la JVM</li> <li>▪ No se conoce bien todas sus capacidades</li> </ul>
VB .NET 2008	<ul style="list-style-type: none"> <li>▪ Nuevas y avanzadas herramientas</li> <li>▪ Creación de sitios web agradables</li> <li>▪ Fácil aprendizaje</li> <li>▪ Nuevas librerías</li> </ul>	<ul style="list-style-type: none"> <li>▪ Licencia propietario</li> <li>▪ No es multiplataforma</li> </ul>

**Tabla 3: Ventajas y Desventajas Lenguajes de Programación**

Con la breve descripción de éstos lenguajes de programación se concluye, tomando en consideración los objetivos que persigue se ha optado por trabajar con JAVA porque este lenguaje ofrece un sin número de ventajas y facilidades para el desarrollo de una aplicación web segura, confiable y agradable para el usuario.

La gran cantidad de librerías con las que cuenta Java así como el fácil manejo que tiene al momento de diseñar aplicaciones web, su seguridad, la convierte en la mejor opción, para el desarrollo de la aplicación, que será publicada mediante un servidor web para su manipulación tanto local como remotamente en cualquier parte que se tenga acceso a internet.

### **3.2.3. SERVIDORES WEB**

#### **3.2.3.1. IIS**

Internet Information Services, no solo es un servidor web sino que ofrece una serie de servicios para computadores con Windows. Al principio fue parte del paquete de opciones que traía Windows NT, pero luego fue incluido en versiones posteriores de Windows como Windows 2000 o Windows Server 2003. En Windows XP Profesional se incluye una versión del IIS pero es limitada, entre los servicios que ofrece el IIS se destaca: FTP<sup>39</sup>, SMTP<sup>40</sup>, NNTP<sup>41</sup> y HTTP/HTTPS<sup>42</sup>.

Todos estos servicios ofrecidos por IIS proporcionan las funciones necesarias para la administración de un servidor web de forma segura es así que cuando una computadora tiene instalado este servicio está en la capacidad de publicar páginas web como en cualquier otro servidor web.

Este servidor web trae consigo varios módulos que le dan la capacidad de procesar distintos tipos de páginas web, como es el caso de ASP, ASP.NET, PHP, CGI<sup>43</sup>, Perl<sup>44</sup>, de esta manera se convierte en un

---

<sup>39</sup> Protocolo de transferencia de archivos.

<sup>40</sup> Protocolo Simple de Transferencia de Correo.

<sup>41</sup> Network News Transporte Protocolo (protocolo para la transferencia de noticias en red).

<sup>42</sup> HTTP Seguro.

<sup>43</sup> El CGI por sus siglas en inglés 'Common Gateway Interface' es de las primeras formas de programación web dinámica.

<sup>44</sup> Perl es un acrónimo de Practical Extracting and Reporting Language, que indica que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.



servidor muy robusto y estable, no solo por soportar todos éstos lenguajes sino porque permite múltiples usuarios conectados simultáneamente.

IIS funciona bajo Windows Server pero de igual forma viene incluido en versiones de Windows XP y Vista pero con funciones reducidas como en número de conexiones, haciendo de esta versión del IIS sea más utilizada para pruebas y diseño.

Al ser Windows un sistema operativo propietario y al traer incluido el IIS se da un costo adicional, contando con la ventaja de tener soporte por parte de Microsoft y de sus usuarios.

#### **3.2.3.2. Apache**

El Servidor Apache es un servidor Web con tecnología Open Source robusto desarrollado por la Apache Software Foundation, es un servidor seguro y confiable, soporta PHP, CGI, Perl, Java, etc., también se puede publicar aplicaciones ASP pero su integración no es segura ni eficiente.

Apache es una herramienta adicional si es instalada en sistemas operativos Windows motivo por el cual, el soporte únicamente es dado por los propios usuarios, por éste motivo se ha convertido en uno de los servidores más populares ocupando el 70% de los sitios web en el mundo.

Principales funcionalidades de Apache:

- Los módulos Apache API
- Apache está disponible en varias plataformas
- Soporta IPv6
- Proporciona respuestas a errores en diversos idiomas

La mayor preocupación que existe con este servidor es que en programas de elevado consumo de procesador y múltiples conexiones se produce su caída.

### 3.2.3.3. Cherokee

Este servidor web open source liberado con una licencia GPL, escrito en lenguaje C, es uno de los servidores que va tomando fuerza en el mercado de la web debido sus características de alto rendimiento, también contando con una fácil instalación y configuración.

Es un servidor multiplataforma actualmente es desarrollado y mantenido por la comunidad de desarrolladores libres, la meta de este servidor es convertirse en un sistema rápido y completamente funcional, y sobre todo liviano a comparación de otros servidores web. Cherokee trata de cubrir las necesidades que tiene Apache brindando soporte en entornos donde existe mucho tráfico, además soporta varios complementos para aumentar su funcionalidad.

En la siguiente tabla, se muestra las ventajas y desventajas de los Servidores Web anteriormente analizados.

SERVIDOR	VENTAJAS	DESVENTAJAS
IIS	<ul style="list-style-type: none"><li>▪ Seguro y confiable</li><li>▪ Ofrece más servicios</li><li>▪ Múltiples usuarios conectados simultáneamente</li><li>▪ Soporte de Microsoft</li></ul>	<ul style="list-style-type: none"><li>▪ Sistema Propietario</li><li>▪ Solo trabaja bajo Windows</li></ul>
APACHE	<ul style="list-style-type: none"><li>▪ Servidor Multiplataforma</li><li>▪ Mas usado en sitios web</li><li>▪ Confiable, seguro, estable</li><li>▪ Open Source</li><li>▪ Fácil instalación</li><li>▪ Respuestas a errores en diversos idiomas</li></ul>	<ul style="list-style-type: none"><li>▪ Gran tamaño</li><li>▪ Caída con múltiples usuarios conectados</li></ul>
CHEROKEE	<ul style="list-style-type: none"><li>▪ Open Source</li><li>▪ Liviano</li><li>▪ Fácil instalación</li><li>▪ Alto rendimiento</li></ul>	<ul style="list-style-type: none"><li>▪ No es muy utilizado</li><li>▪ Se está en la espera de su máximo potencial</li></ul>

Tabla 4: Ventajas y desventajas Servidores Web

Al revisar y comparar las distintas utilidades que ofrecen estos Servidores Web y de igual forma viendo sus debilidades, pero sobre todo sin perder la perspectiva que persigue este proyecto de tesis, por funcionalidad,

seguridad, grado de estabilidad, costos, popularidad en el mercado actual de la web, entendimiento con el lenguaje de programación y base de datos previamente seleccionados: se concluyó que el Servidor Apache es la mejor alternativa que se adapta a los requerimientos que tiene el sistema.

Es así que se adoptó Apache Tomcat como servidor web en su versión 6.0.26 para realizar el desarrollo, implementación y publicación de nuestro sitio web.

### **3.2.4. SERVIDORES DE CORREO WEB**

#### **3.2.4.1. Hotmail**

Windows Live Hotmail, antiguamente MSN Hotmail o simplemente como Hotmail, es uno de los servidores de correo electrónico gratuito de propiedad de Microsoft, fue uno de los primeros proveedores de correo gratuitos.

La versión actual, Windows Live Hotmail, fue lanzada oficialmente en todo el mundo el 2007, Hotmail ofrece 5 GB de almacenamiento, cuenta con más de 270 millones de usuarios y se encuentra en más de 36 países.

Características de Windows Live Hotmail.

- Utiliza tecnología Ajax.
- Búsqueda avanzada.
- Organización de correos en carpetas.
- Análisis de virus y filtro de spam<sup>45</sup> en correos.
- Soporta múltiples direcciones.
- Corrector ortográfico en la redacción de correos.

---

<sup>45</sup> Correo basura son mensajes no solicitados, no deseados o de remitente desconocido, en su mayoría de tipo publicitario.

### 3.2.4.2. GMAIL

GMAIL es un servicio gratuito de correo electrónico ofrecido por Google con capacidad prácticamente ilimitada, en la actualidad es la única empresa en brindar los servicios de correo con estas características.

Entre las tantas utilidades que ofrece GMAIL una de las más destacadas es la posibilidad de trabajar con los protocolos POP3<sup>46</sup> e IMAP<sup>47</sup>.

Utiliza tecnología AJAX, siendo pioneros en emplearla dentro de éste tipo de servicios, aunque también se encuentra en una interfaz HTML con CSS<sup>48</sup> utilizada más en navegadores antiguos o no compatibles.

Los servicios que ofrece GMAIL se destacan ante sus competidores por ser muy innovadores y de suma utilidad al momento de revisar un correo, es así que se realiza búsquedas avanzadas, se cuenta con un mecanismo de etiquetado de mensajes que elimina prácticamente el almacenamiento en carpetas (haciendo mucho caso a su eslogan “Busca, no ordenes”), proporciona un sofisticado filtro de spam que los elimina instantáneamente, también realiza traducciones de los correos entrantes a otros idiomas

GMAIL también está disponible en dispositivos móviles, aunque no con las mismas características del servicio tradicional que se ofrece.

---

<sup>46</sup> Protocolo 3 de Correo es un protocolo estándar para recibir mensajes de e-mail. Los mensajes de e-mails enviados a un servidor, son almacenados por el servidor Pop3.

<sup>47</sup> Internet Message Access Protocol. Es un protocolo de red que permite acceder a mensajes de correo electrónico que están almacenados en un servidor.

<sup>48</sup> Cascading Style Sheets u Hojas de Estilo en Cascada es la tecnología desarrollada con el fin de separar la estructura de la presentación.

En la siguiente tabla, se muestra las ventajas y desventajas de los Servidores de Correo Web analizados.

WEBMAIL	VENTAJAS	DESVENTAJAS
HOTMAIL	<ul style="list-style-type: none"> <li>▪ Utiliza Ajax</li> <li>▪ Manejo de carpetas</li> <li>▪ Soporta múltiples direcciones</li> <li>▪ Filtro Spam y antivirus</li> </ul>	<ul style="list-style-type: none"> <li>▪ Correo desordenado y sin etiquetas</li> </ul>
GMAIL	<ul style="list-style-type: none"> <li>▪ Utiliza Ajax</li> <li>▪ Almacenamiento prácticamente ilimitado</li> <li>▪ Confiable, seguro</li> <li>▪ Etiquetado de mensajes</li> <li>▪ Filtro Spam y antivirus</li> <li>▪ Traducción de correos</li> </ul>	<ul style="list-style-type: none"> <li>▪ Poco de dificultad en crear una cuenta</li> </ul>

**Tabla 5: Ventajas y desventajas servidores de Correo Web**

Evidentemente al momento de diferenciar y escoger entre los Servidores de Correo Web gratuitos se observa que no existe mucha diferencia entre ellos y los servicios que ofrecen, sino es cuestión de gustos, costumbre o de cómo cada usuario se sienta mejor, porque de otra forma en servicios y tecnología no existe mucha diferencia entre ellos.

Es por esto que se ha escogido a GMAIL como Servidor de Correo Web para que trabaje conjuntamente con la aplicación, también por razones de compatibilidad con el lenguaje de programación escogido (Java), que entre sus múltiples librerías contiene dos que enlazan directamente a Java con GMAIL. Y otra motivo es que GMAIL lleva un control de correos de forma más ordenada y etiquetada a diferencia de otros servidores.

### **3.2.5. LENGUAJES DE PROGRAMACIÓN DE MICROCONTROLADORES**

Existen varios lenguajes de programación así como también compiladores de Microcontroladores unos más avanzados y con más funciones, pero todos tendrán la misma finalidad que es controlar los Microcontroladores de la manera que requiera el sistema. Los tres lenguajes de programación que se analizará a continuación, son los siguientes:

#### **3.2.5.1. Lenguaje BASIC**

##### **Ventajas:**

- Lenguaje de fácil manejo muy simple y con instrucciones fácilmente legibles hasta para programadores no muy avanzados.

##### **Desventajas:**

- Complicado manejo de interrupciones simultáneas en este lenguaje.
- Tiene limitaciones al genera el archivo .hex, por este motivo no optimiza el tamaño del programa con relación a la memoria del PIC.
- Únicamente trabaja bajo ambiente Windows.

#### **3.2.5.2. Lenguaje C**

##### **Ventajas:**

- Lenguaje de alto nivel más cercano a la máquina.
- Construcción de rutinas matemáticas fácilmente.
- Compatible con Ensamblador sobre todo en PICs de gama alta.
- Creación de macros con este lenguaje, para después simplificar el código.

**Desventajas:**

- La compilación de los programas resulta en muchos casos muy extensos y pesados, es por esto que se debe tener muy en cuenta la capacidad de memoria del PIC que se está programando.

**3.2.5.3. Lenguaje Ensamblador (assembler)****Ventajas:**

- Es el lenguaje de bajo nivel natural de los PICs tanto para gama baja, media o alta.
- Eficiente aprovechamiento de los recursos del PIC.
- También permite la creación de macros.
- Controla los tiempos y los registros bit a bit.
- Controla muy bien las interrupciones simultáneas.
- Al generar el archivo .hex éste es completamente optimizado.

**Desventajas:**

- Requiere cierta experiencia de programación.
- Código extenso.

Para el manejo de los Microcontroladores se ha escogido el Lenguaje Basic, debido a que posee estructuras, sentencias e instrucciones muy legibles y de fácil manipulación al momento de programar, por tales motivos mediante Basic se generará el código de programación que controlará a los Microcontroladores de cada módulo.

### **3.2.6. HERRAMIENTAS DE DESARROLLO JAVA**

Existen varias herramientas o editores en las que se programa con Java, unas creadas por la misma Sun Microsystems y otras desarrolladas por diversas empresas entre las más populares se tienen:

#### **3.2.6.1. Eclipse**

Plataforma Open Source<sup>49</sup> de Desarrollo para aplicaciones basadas en Java y otros lenguajes.

#### **3.2.6.2. Netbeans**

Herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas java, agradable y fácil manipulación de su entorno gráfico.

#### **3.2.6.3. JCreator Pro**

Un potente entorno de desarrollo para crear aplicaciones Java, mucho más completo que editores similares.

#### **3.2.6.4. BlueJ**

Es un completo entorno Java desarrollado como una solución especialmente enfocada a la enseñanza.

#### **3.2.6.5. AutoJava**

Es una herramienta para crear aplicaciones programadas en Java con la ventaja de que así pueden ser utilizados en todos aquellos sistemas operativos que tengan Java instalado.

Al existir un sin número de herramientas en las que se programa Java se ve la necesidad de escoger una que posea mayores ventajas y que sobresalga ante el resto.

---

<sup>49</sup> Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente.



Se ha escogido la herramienta Netbeans en su versión 6.5.1 manejando aplicaciones Web con el Framework Visual Web Java Server Faces, debido a que posee un entorno visual de fácil manejo además de un gran número de componentes que facilitan la programación Web.

### **3.2.7. HERRAMIENTAS DE DESARROLLO BASIC PARA MICROCONTROLADORES**

#### **3.2.7.1. MikroBasic**

Es una poderosa herramienta para programar Microcontroladores mediante el lenguaje Basic.

##### **Características:**

- mikrolCD – Depurador en Circuito.
- Gestor de proyectos: permite a los usuarios gestionar múltiples proyectos.
- Explorador de Código: permite supervisar las variables, funciones, procedimientos que se están usando.
- Administrador de Librerías: permite ver las librerías que se están utilizando y las librerías que son almacenadas al momento de compilar el programa.
- Asistente de código: Ahorra tiempo al momento de escribir código, basta con presionar las teclas Ctrl + Espacio y aparece la lista de identificadores.

#### **3.2.7.2. Microcode Studio**

Es una herramienta muy potente para la programación de Microcontroladores en lenguaje Basic diseñada específicamente para MicroEngineering Labs PBP y PBP PRO.

El editor principal de esta herramienta proporciona la sintaxis completa y resalta el código con ayuda contextual.

Tiene un explorador de código que permite incluir archivos, constantes, variables, modificadores y etiquetas que están dentro del código fuente que se genera, de igual forma posee características de búsqueda y remplazo.

Contiene una ventana de mensajes de error basta con hacer clic sobre el error y automáticamente el Microcode Studio te llevará a la línea de código donde se encuentra el error, también tiene una ventana de comunicaciones, que permite depurar y ver la salida del microcontrolador.

Para la programación de los Microcontroladores se escogió las dos herramientas analizadas anteriormente, MikroBasic V 7.0.0.2 con la cual será programado el microcontrolador que controlará el circuito del GLCD que requiere un mayor grado de dificultad al momento de manejar las imágenes (cuenta con GLCD Bit Map), así como también facilita el uso de funciones que son necesarias para manejar éste dispositivo.

Por otro lado será también utilizado el Microcode Studio V3.0.0.5 con el cual serán programados los Microcontroladores que tienen los demás circuitos del sistema.

### **3.2.8. HERRAMIENTAS DE DESARROLLO APLICACIONES MÓVILES BLUETOOTH**

Para el desarrollo de la aplicación Bluetooth móvil se tomó en cuenta dos herramientas de desarrollo que continuación serán descritas:

#### **3.2.8.1. Mobile Processing**

Es un entorno para dispositivos móviles que permite desarrollar de forma sencilla aplicaciones y prototipos. Es una parte del proyecto Processing que originalmente fue diseñado para crear entornos Web.

En esta herramienta se escribe código de una manera sencilla y de forma más intuitiva a diferencia de otras herramientas Java.

Este IDE posee una licencia de código abierto que permite acceder al programa y a su código de forma gratuita, al ser libre se convierte también en una herramienta multiplataforma.

Al ser utilizado en Windows se debe manejar el Sun Wireless Toolkit distribuido gratuitamente por Sun Microsystems, para la compilación y simulación de las aplicaciones que se creen en Mobile Processing.

#### **3.2.8.2. J2ME Wireless Toolkit**

Es un sencillo IDE para el desarrollo de MIDlets es decir, programas en lenguaje de programación Java para dispositivos embebidos generalmente son juegos o aplicaciones que corren en un dispositivo móvil.

Esta herramienta permite compilar, pre verificar y ejecutar el código en emuladores de teléfonos móviles como Nokia, Sony Ericsson etc.

Previo al análisis de estas herramientas desarrolladoras de aplicaciones móviles se escogió a Mobile Processing 007 para el desarrollo de la aplicación Bluetooth del proyecto, provee de una mayor sencillez al momento de programar este tipo de aplicaciones y es muy intuitivo al momento de generar código.

#### **3.2.9. SIMULADORES DE CIRCUITOS ELECTRÓNICOS**

En su gran mayoría todos los simuladores de circuitos electrónicos son con licencia propietaria debido a que su desarrollo conlleva a que los productos terminados sean comercializados, a excepción de algunos simuladores que son orientados mas para el aprendizaje en instituciones educativas que cuentan con licencias gratuitas como es el caso de LogicCircuit, ElectroWin, MultiSIM, etc., con la gran desventaja de no tener funciones avanzadas al momento de diseñar circuitos y en si son versiones muy limitadas.

Esta desventaja no presentan los simuladores más sofisticados como es el caso de Protel, Proteus, OrCad (entre los más conocidos y usados) que traen una extensa lista de librerías de dispositivos electrónicos y hasta de microprocesadores, convirtiéndole al diseño de circuitos en una tarea más sencilla, rápida, inteligente y óptima de ejecutar.

Tomando en cuenta algunas ventajas que serán descritas más adelante, se ha tomado como herramienta de diseño de Circuitos a Proteus en su versión 7.5 sp3, en el cual se ha encontrado un fácil diseño de circuitos, rápida simulación de circuitos electrónicos con Microcontroladores, altas prestaciones, es el software más popular para la simulación de Microcontroladores PIC, posee una herramienta poderosa para el diseño de placas de circuito impreso y en especial su gran cantidad de librerías de dispositivos electrónicos.

#### **3.2.9.1. Proteus**

Proteus desarrollado por Labcenter Electronics consta de sus dos programas principales que son: Ares e Isis.

- **Ares:** Advanced Routing and Editing Software (Software de Edición y Ruteo Avanzado); es la herramienta que ocupa Proteus para el enrutado, ubicación y edición de componentes, para la fabricación de placas de circuito impreso, que permite editar las 2 capas que tiene un circuito: capa superficial (Top Copper), y de soldadura (Bottom Copper).
- **ISIS:** Intelligent Schematic Input System (Sistema de Enrutado de Esquemas Inteligente) con este programa se diseña el plano eléctrico del circuito, y éste puede ser simulado en tiempo real, mediante el módulo VSM (Virtual System Modeling o Sistema Virtual de Modelado).

### **3.3. ANÁLISIS DE HARDWARE**

#### **3.3.1. MÓDULO GSM**

GSM o Sistema Global para las Comunicaciones Móviles es un sistema estándar, utilizado para la comunicación por medio de teléfonos móviles que cuenten con tecnología digital, esto también hace posible conectar el teléfono móvil con un computador para realizar tareas como: envío y recepción de mensajes, fax, navegar por internet, acceso a redes LAN, entre otras, pero una de las más utilizadas e innovadoras, el Servicio de Mensajes Cortos o SMS.

En la actualidad más se habla de 3G, que no es más que la evolución del GSM, ésta difiere a sus versiones anteriores en la velocidad de transmisión, arquitectura, protocolos de radio utilizados y servicios ofrecidos.

3G maneja velocidades de hasta 3 megas por segundo para la navegación por internet permitiendo una mayor calidad en los servicios que ofrece es así que el envío de pequeños archivos, fotos, mensajes de voz, timbres o sonidos, video en línea, escuchar música, se realiza sin ningún problema mediante ésta tecnología, y en general la calidad y capacidad es mucho mayor.

Otro objetivo que se ha conseguido con GSM es lograr que éste servicio no quede limitado a un solo país o ciertas zonas específicas sino que traspase las fronteras es así que ahora tranquilamente se tiene un mismo número y una comunicación entre más de 150 países alrededor del mundo porque ésta es una tecnología satelital.

##### **3.3.1.1. Servicios GSM**

Los siguientes servicios son los más importantes con los que cuentan los dispositivos GSM.

#### **3.3.1.1.1. SMS**

Es un servicio de mensajes cortos disponible en teléfonos móviles que permite el envío de mensajes. En su origen SMS fue diseñado como parte del estándar de telefonía móvil digital GSM, pero actualmente se encuentra presente en una variedad de redes, y es usado en gran cantidad y no tiene un costo elevado.

Cuando se envía y recibe SMS mediante teléfonos móviles siempre se generan parámetros que identifican el mensaje tales como: fecha de envío, validez del mensaje, número de teléfono remitente y destinatario.

#### **3.3.1.1.2. Mensajes multimedia**

MMS o sistema de mensajería multimedia es un estándar de mensajería que tiene los teléfonos móviles para enviar y recibir contenidos multimedia, que incorporan sonido, video, fotos. La mensajería multimedia se envía a cuentas de correo electrónico, ampliando las posibilidades de la comunicación móvil, pudiendo publicar nuestras fotografías digitales o actuar en weblogs sin mediación de un ordenador. Existe un límite de cada mensaje multimedia suele ser de 100 o 300 KB, dependiendo del teléfono que se use, y no mas por limitaciones del protocolo. El costo de envío de estos mensajes es superior al de SMS dependiendo de la cantidad de información enviada.

Por lo descrito anteriormente en especial el servicio de SMS que ofrece GSM será utilizado en el proyecto de tesis mediante un teléfono Nokia 3220b, que servirá como módulo GSM para la transmisión y recepción de mensajes SMS.

Mediante éstos mensajes se gestiona el sistema, esto se conseguirá al conectar el teléfono móvil GSM directamente al servidor web y éste a su vez estará comunicado a la red Zigbee, para realizar las funciones de

módem GSM en el servidor y por ende todos los dispositivos que se encuentren conectados a esta red.

**Nota:** Para ver las especificaciones técnicas completas del teléfono Nokia 3220b ver en el Anexo 3.

### **3.3.2. MÓDULOS XBEE**

Una de las empresas líderes en fabricación de módulos Xbee es DIGI que se distribuye en el mundo con oficinas en América del Norte, Europa y Asia con aproximadamente 262 distribuidores en más de 70 países, convirtiéndola en la empresa más grande en fabricación de módulos Zigbee se refiere, y con un sin número dispositivos electrónicos que adoptan también ésta tecnología.

Otra empresa que a lo largo de estos últimos años se ha visto comprometida con las necesidades que exige el mercado en relación a módulos Xbee y redes inalámbricas se refiere es Microchip que ha lanzado sus gama de nuevos productos basados en Zigbee, y se espera que sea una de las competidoras de DIGI a pesar de que ésta última ya lleva una gran ventaja.

Otra factor a tomar en cuenta son las dos series que ofrecen los módulos Xbee serie 1 y 2 cada una con sus propias características, cabe señalar que estas dos series no son compatibles entre ellas así que se recomienda tener cuidado al momento de utilizarlas.

En la siguiente tabla, se definen las características de los módulos Xbee Serie 1 y 2.

	Xbee Series 1	Xbee Series 2
Alcance	100 ft. (30m)	133 ft. (40m)
Potencia de salida	1 mW (0dbm)	2 mW (+3dbm)
Transmisión	250 Kbps	250 Kbps
Voltaje de Alimentación	2.8 - 3.4 V	2.8 - 3.6 V
Consumo Rx	45mA	40mA
Consumo Tx	50mA	40mA
Frecuencia	ISM 2.4 GHz	ISM 2.4 GHz
Dimensiones	0.0960" x 1.087"	0.0960" x 1.087"
Temperatura de Funcionamiento	-40 to 85 C	-40 to 85 C
Tipos de Antena	Chip, Integrated Whip, U.FL	Chip, Integrated Whip, U.FL <sup>50</sup> , RPSMA <sup>51</sup>
Topologías	Point to point, Star	Point to point, Star, Mesh

**Tabla 6: Diferencias entre Módulos Xbee Serie 1 y 2**

Después de esta breve referencia de dos grandes empresas desarrolladoras y distribuidoras de módulos Zigbee se pudo notar que DIGI al ser un líder en fabricación de éstos productos se convierte en la mejor opción a seguir, motivo por el cual se ha escogido los productos ofrecido por DIGI tomando en cuenta su liderazgo, disponibilidad en el mercado, soporte y costo.

En éste proyecto de tesis se empleará 5 módulos Xbee Serie2 (debido a que los Xbee serie1 no se configuran en topología Malla) marca DIGI para la implementación de la red topología MESH/MALLA con características técnicas que serán descritas a continuación:

<sup>50</sup> U. FL, un conector coaxial de RF en miniatura para señales de alta frecuencia fabricados por Hirose Electric Group.

<sup>51</sup> RPSMA Reverse-polarity SubMiniature version A (conectores de polaridad inversa Sub Miniatura versión A).



**Modelo:** XB24-Z7WIT-004

**Compatibilidad:** Solo con otros módulos Xbee Serie 2 (no es compatible con la serie Xbee 1)

**Alcance:** 40 m para interiores y 120 m en el exterior (en exterior siempre y cuando se cuente con línea de vista)

**Configuración:** Comandos API o AT, local o por el aire

10 E / S digitales y (4) entradas ADC

**Banda de frecuencia:** 2.4 GHz aceptada en todo el mundo

**Potencia de transmisión:** 1,25 mW (1dBm)

**Antena:** Alambre

**Velocidad de Transmisión:** 250 kbps / hasta 1 Mbps

**Topología:** Malla

**ID:** PAN ID, MAC IEEE de 64 bits

**Voltaje de alimentación:** 2,1 a 3,6 VCC

**Nota:** Para ver las Dimensiones y especificaciones técnicas de los Módulos Xbee ver en el Anexo 4.



Figura 4: Fotografía Módulo Xbee Serie2

### **3.3.3. ANÁLISIS GLCD**

Sin duda en la actualidad se manejan muchos los dispositivos móviles como celulares, PDA, computadores portátiles etc., que en algunos casos son más sofisticados que otros, pero las grandes empresas desarrolladoras de estos dispositivos están innovando cada vez más, es muy fácil encontrar equipos Touch o táctiles que no son otra cosa que dispositivos equipados con un GLCD capaz de manejar imágenes o iconos esparcidos a lo largo y ancho de la pantalla que al ser presionados cumplen con determinada función, esto se hizo para tener mejor control y eliminar en gran cantidad de botones que se requiere para que un dispositivo pueda ser manipulado.

Es por esto que, el uso de pantallas táctiles está ganando mucho espacio en el mercado del los equipos móviles como ejemplos se menciona los siguientes:

Teléfono Nokia 5800 Xpress, computadora HP Pavilion TX2010es, teléfono huawei u7515, etc., de igual manera en este proyecto de tesis se vio la factibilidad de implementar un control inalámbrico de éstas características para el manejo de el sistema de una manera innovadora y muy agradable.

#### **3.3.3.1. Tipos de GLCD Táctiles**

Se distinguen varios tipos de GLCD táctiles los cuales son:

##### **3.3.3.1.1. Infrarrojos**

Este tipo de pantallas contiene en uno de sus bordes un emisor infrarrojo y en otro borde un receptor infrarrojo, de igual manera una matriz infrarroja que al presionarla se interrumpe el haz de luz vertical y otro horizontal. El computador del GLCD detecta cuales rayos fueron interrumpido y genera la posición.

La desventaja de este tipo de GLCD es su alta sensibilidad que produce falsas posiciones, y también es su elevado costo.

### **3.3.3.1.2. Resistivas**

Está formada por dos capas de material conductor que tienen cierta resistencia a la corriente eléctrica, al ser presionada se crea contacto entre estas dos capas y se mide la resistencia causada, mediante este proceso, se conoce exactamente cuál fue el punto presionado.

Este tipo de pantalla es sensible a la luz ultravioleta haciendo que se deteriore rápidamente, su precio es económico y son muy confiables.

### **3.3.3.1.3. Pantallas táctiles capacitivas**

Este tipo de pantallas contiene una capa conductora dentro del cristal del propio tubo y aplica una tensión sobre las cuatro esquinas de la pantalla. También posee una capa que captura las cargas y cuando la pantalla es tocada con algún otro material se genera un decremento de energía.

El software que poseen estas pantallas hace posible que se calcule la carga entre cada esquina para conocer con exactitud cuál fue el punto de contacto.

La mayor ventaja de este tipo de pantallas es tener una imagen más clara, esto se debe a que tiene menos capas sobre el monitor.

Tomando en consideración costos, tamaños, memoria, marcas, controladores que ocupa, tipos de pantallas se hizo una comparativa de cuál de todos los GLCD táctil sirven adecuadamente para el desarrollo de éste proyecto de tesis, se encontró GLCDs con controladores Samsung KS0107, Samsung KS0108 o el Toshiba T6963C, pero se optó por éste último con una resolución de 240 x 128 pixeles de tipo Resistivo, adecuado para el manejo de imágenes, con una resolución alta que no distorsionará ni hará difícil la visualización, un factor que influyo mucho para escoger este modelo GLCD fue la disponibilidad, debido a que no se contaba con otros modelos ni con otro tipo de resolución (obviamente mayor).

**Nota:** Para ver el Datasheet del GLCD utilizado ver el Anexo 5.



Figura 5: Fotografía GLCD

### 3.3.4. ANÁLISIS DE SENSORES

#### 3.3.4.1. Sensor Magnético

Existen una gama muy extensa de éstos dispositivos unos más sofisticados que otros que al momento de su elección se convierte en una tarea muy tediosa de realizar tomando en cuenta que todos estos se basan en el mismo principio magneto resistivo, motivo por el cual se ha escogido uno de los más sencillos pero no por esto menos eficientes, sino que éstos tranquilamente cumplirán con la tarea que van a realizar.

En éste proyecto serán requeridos 4 sensores magnéticos (2 en cerraduras y 2 en el sistema de cortinas) con características que se detallan a continuación:

- Normalmente Abiertos.
- Tornillos de fijación en ambos extremos (soporte electrónico y soporte magnético).
- Presentación: Blanco.
- Tipo de montaje: Plástico.
- Dimensiones: 3.3 cm x 1.3 cm.
- Cinta adhesiva en ambos soportes.

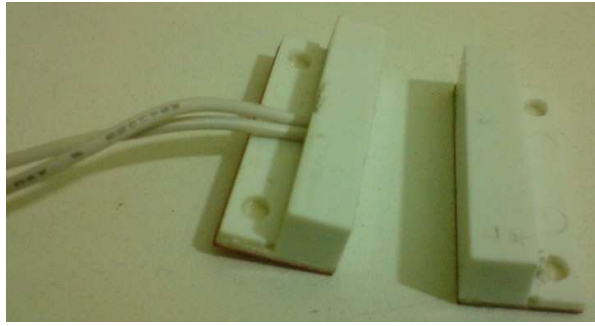


Figura 6: Fotografía Sensor Magnético

#### 3.3.4.2. Sensor Fotovoltaico

Se va a utilizar un LDR sencillo que cumpla con su función base o en otras palabras un semiconductor fotosensible (es decir, cuando más luz exista menos resistencia habrá) con las siguientes características técnicas:

- 1 mΩ, o más, en la oscuridad y 100 Ω con luz brillante.
- Es sensible a una gama de frecuencias, incluyendo infrarrojo (IR), luz visible, y ultravioleta (UV).
- Tiempo de respuesta: En el orden de una décima de segundo.
- Alimentación: Corriente alterna.
- Material de construcción: Sulfuro de cadmio.

**Nota:** Para ver el Datasheet del LDR utilizado ver el Anexo 6.



Figura 7: Fotografía LDR

### 3.3.4.3. Sensores de Movimiento

Existen varios tipos sensores capaces de detectar movimiento y como se describió previamente en el capítulo 2 se puso en consideración los 2 tipos de detectores más conocidos y utilizados, que son los detectores Infrarrojos Pasivos de Movimiento y los de microondas, cada uno presenta soluciones diferentes y diversas, pero para motivos de mayor seguridad y confiabilidad se ha optado por los detectores pasivos infrarrojos de movimiento que ofrecen un mejor grado de prevención contra falsas alarmas.

Serán utilizados 2 detectores de movimiento de infrarrojo pasivo con las siguientes características técnicas:

- **Modelo:** TS-6511.
- **Voltaje:** 12 VDC.
- **Período de calentamiento:** 30 segundos.
- **Sensibilidad:** Baja, Media, Normal.
- **Altura de montaje:** 2.2 m.
- **Patrones de detección:**
  - Lente estándar - Rango 15 m - Angulo de vista 90° - zonas detectadas 48.
  - Lente cortina – Rango 12 metros – zonas detectadas 11.
  - Lente de largo alcance – Rango 30 m – zonas detectadas 10.

**Nota:** Para ver las especificaciones técnicas del detector de movimiento ver el Anexo 7.



Figura 8: Fotografía Detector de Movimiento

#### 3.3.4.4. Detector de Humo

Dos tipos de sensores de humo fueron tomados en cuenta: uno es el sensor de humo por efecto fotoeléctrico y el otro sensor de humo por ionización características que fueron ya analizadas con anterioridad en el capítulo 2, guiados en los beneficios ofrecidos por los detectores de humo por ionización se optó por utilizarlos, más por su alto grado de sensibilidad y porque son capaces de detectar incendios de desarrollo rápido es decir, cuando existe la presencia de fuego.

El detector de humo utilizado está descrito a continuación, con sus especificaciones técnicas:

- **Marca:** Kidde.
- **Modelo:** i9030.
- **Técnica utilizada:** Ionización.
- **Fuente de energía:** Batería de 9 voltios.
- **Audio de la alarma:** 85db a 3 m.
- **Peso:** 1 libra.
- **Rango de temperatura:** 4 ° C y 38 ° C.
- **Humedad de funcionamiento:** 85%.
- **En caso de humo:** 3 pitidos de forma repetitiva.

- **Aviso de estado de la batería:** 1 pitido cada 30 a 40 segundos por un período de 7 días.



Figura 9: Fotografía Detector de Humo

**Nota:** El Datasheet del detector de humo se encuentra en el Anexo 8.

#### 3.3.4.5. Detector de Gas

Existen una infinidad de gases éstos pueden ser inflamables, tóxico, explosivo, oxígeno, nitrógeno etc., y de la misma forma se encontró varios sensores que detectan ciertos tipos de gases y otros no.

Analizando los requerimientos que éste sistema persigue se observa que será implementado en domicilios donde la mayor preocupación que existe es la existencia de una fuga de gas LP (licuado de petróleo) compuesto mayormente de propano y butano (presentes en el petróleo y gas natural), es así que al escoger este tipo de sensor se enfocó en uno que pueda detectar en especial éste tipo de gas.

Considerando el costo, función que va a tener, y utilidad se ha optado por el sensor MQ-6 el cual tiene las siguientes especificaciones técnicas:

- Buena sensibilidad al gas combustible en amplia gama.
- Tiempo de respuesta rápido.
- La salida del sensor tiene una resistencia analógica.



- Alimentación del calentador 5 voltios.
- Detecta concentraciones de hasta 200 hasta las 10000ppm.
- Poca sensibilidad al alcohol y al smoke.
- Larga vida.
- Simple manejo del circuito.
- Gases que detecta: Isobutano, butano, GLP.
- Fácil funcionamiento: Simplemente hay que alimentar el calentador con 5V agregar una resistencia de carga y conectar la salida a un ADC.

**Nota:** Las especificaciones técnicas del detector de gas son mostradas en el Anexo 9.



Figura 10: Fotografía Detector de Gas GLP

### 3.3.5. ANÁLISIS DE ACTUADORES

#### 3.3.5.1. Sirena

En todo sistema de seguridad se cuenta con algún tipo de dispositivo que de una alerta cuando está ocurriendo algo o se esté manipulando de forma equivocada, y la mejor opción es una alerta auditiva que atraiga nuestra atención y ponga en sobre aviso de que algún acontecimiento está sucediendo, para de esta manera acudir a solucionarlo.

Por este motivo no podía faltar uno de éstos dispositivos en el sistema, el cual dará un aviso cuando algo anormal ocurra o interrumpa el correcto funcionamiento de nuestro sistema.

Se ha optado por una sencilla pero eficiente sirena de viento que alertará al usuario del sistema y de igual manera creará una alerta general alrededor del lugar donde se encuentre el sistema implementado.

Especificaciones básicas de la sirena que será implementada:

- Sirena de viento.
- Potencia de sonido: 60db aproximadamente.
- Dimensiones: Altura 9cm – Largo 8cm.
- Alimentación: 12 voltios DC.
- Soporte: 2 tornillos laterales.



Figura 11: Fotografía Bocina

### **3.3.5.2. Cámara IP**

Siempre pensando en los objetivos que persigue éste proyecto de tesis y al hacer un análisis de costos, utilidad que se le va dar, se ha podido concluir que este sistema requiere de un dispositivo que no sea extremadamente sofisticado y con funciones excesivamente extensas, sino por lo contrario un dispositivo básico con funciones específicas que serán utilizadas y no desaprovechadas como ocurriría en el caso de los dispositivos más avanzados.

Lo que se requiere en éste sistema es de una cámara IP que envíe imágenes a través del servidor web en forma continua, eficiente, que sea de fácil configuración, para ver lo que acontece dentro del lugar vigilado y así tener una idea clara de los acontecimientos que puedan darse al interior de éste de manera instantánea (en tiempo real).

Se ha elegido un modelo de cámara IP relativamente sencillo pero que en la práctica cumplirá muy bien con la tarea designada.

Características técnicas cámara IP D- link modelo DCS-910:

- CPU interno.
- Servidor web incorporado.
- Acceso remoto en cualquier momento.
- Instalación simple.
- Configuración intuitiva y basada en web.
- Red cableada Ethernet/Fast Ethernet.
- Sensor permite captura de video en entornos de baja luminosidad.
- Memoria flash de 8MB.
- Soporte de Balance Automático.
- Distancia focal 4.57mm.
- Zoom digital hasta 4X.
- Soporta compresión MJPEG.
- Soporta JPEG para imágenes fijas.
- Soporta protocolos:
  - IPV4, ARP, TCP, UDP, ICMP.
  - DHCP, NTP, DNS, DDNS, SMTP, FTP, HTTP Server.
- Multiplataforma.
- Temperatura: 0 a 40° C.
- Dimensiones: 71.9 x 110 x 37 mm.
- Alimentación de energía: input 100-240 VAC – output 5 VDC, 2.5.

**Nota:** La ficha técnica de la cámara IP se encuentra en el Anexo 10.



Figura 12: Fotografía Cámara IP

#### 3.3.5.3. Modular de simulación de apertura y cierra de cortinas

Éste proyecto de tesis contempla el control de cierre y apertura de cortinas de forma automática, por esta razón se ve la necesidad de crear un modular que simule ésta tarea, para la implementación y puesta en marcha se ha tomado en cuenta los siguientes requerimientos: espacio, costo, utilidad y funcionalidad, es por éstos detalles que se ha hecho un armazón sencillo y básico que simule de forma práctica el funcionamiento de una cortina eléctrica.

Los requerimientos que se contemplaron para éste armazón son los siguientes:

- Poleas
- Motor
- Cuerda
- Cortinas pequeñas
- Tornillos
- Rueda

**Nota:** El Diagrama del modular de simulación de cortina eléctrica es mostrado en el Anexo 11.



Figura 13: Fotografía Prototipo Cortina Eléctrica

#### **3.3.5.4. Inversor de Giro de un Motor**

Para la puesta en marcha del motor que simulará la apertura y cierre de la cortina que se definió anteriormente se requiere de un inversor de giro para controlar el motor que tiene un alto amperaje, es por esto que se requiere de un Puente H ya sea de relés o transistores y que controle el encendido, apagado, izquierda y derecha.

##### **3.3.5.4.1. Puente H con Transistores**

Un Puente H es un circuito electrónico que permite girar un motor eléctrico en ambos sentidos (avance y retroceso), son utilizados principalmente como convertidores de potencia.

Se puede construir un Puente H con transistores como por ejemplo con: TIPS 125, TIPS 120, 2N2222A.

##### **3.3.5.4.2. Inverso de Giro con Relés**

Aquí se aplica la misma teoría que en un Puente H con transistores, es decir, el objetivo es conseguir que un motor eléctrico gire en ambos sentidos, pero esta vez con la ayuda de Relés, que son dispositivos más económicos y que den el mismo resultado.

Al analizar estas dos maneras de crear un inversor de giro se optó por un inversor que ocupe relés y que pueda ser usado con facilidad e ingenio, sin utilizar materiales muy costosos y de fácil disponibilidad como son los relés.

### **3.3.6. ANÁLISIS DE MICROCONTROLADORES**

Al hablar y analizar los Microcontroladores se está expuesto a tener un sin número de opciones para escoger, es así que sólo con tener claro cuáles son los alcances que tiene el sistema se verá cuál es la mejor opción que se adapta y cumple con los requerimientos necesarios para conseguir un sistema óptimo, seguro, estable y sin fallas.

En el mercado se encuentran diferentes marcas de Microcontroladores como Motorola, Siemens, Atmel con sus famosos AVR's, y uno de los más populares y por ende más usados los PIC's de Microchip Technology, estos últimos combinan una alta calidad, bajo costo y excelente rendimiento, poniendo en consideración la fácil adquisición, costo, desempeño, soporte que ofrecen los Microcontroladores PIC se ha considerado en adoptar éstos, como parte del sistema, a continuación, se hará un breve análisis de cual microcontrolador se utilizará:

#### **Gamas existentes de Microcontroladores PIC's:**

Gama Baja o básica Microcontroladores 16C5X

Son PICS con una serie de recursos limitados, se encuentran con 18 o 28 patitas y se alimenta con una tensión de 2.5v., son ideales para aplicaciones que funcionan con pilas, poseen 33 instrucciones con formato de 12 bits, no admite interrupción y la pila tiene dos niveles.

- **Gama Media Microcontroladores 16FXXX.**

Aquí se encuentra un completo y variado grupo de PICs que abarcan los modelos de 18 a 68 patitas, que controlan varios periféricos.

- **Gama Alta Microcontroladores 17CXXX.**

Maneja hasta 58 instrucciones de 16 bits, contiene un potente sistema de interrupciones, incluye variados controladores de periféricos, puertas de comunicación serie y paralelo, pueden ser ampliarlos con hardware externo.

- **Gama Mejorada Microcontroladores 18FXXX con instrucciones de 16 bits.**

La gama mejorada de los PICS fueron diseñados para soportar aplicaciones avanzadas de automatización. Cuentan con una gran velocidad (40 MHz) y un gran rendimiento.

Al hacer una comparación con los Microcontroladores PICs de gama Media y Alta se pudo notar que existe una marcada diferencia, como es la mayor capacidad de memoria Flash, más módulos internos, más pines, y su set de instrucciones permite hacer programas en menos espacio. Un punto muy importante de destacar es que entre los PICs 16FXXX y los 18FXXX no existe mucha diferencia de costos y en algunos casos es menor, por esto también son más cotizados porque al mismo precio se tiene mayor potencia en los PICs de gama Alta.

En este sistema se hará uso de 4 PICs 18F452 (Gama Alta) que estarán distribuidos en los diferentes circuitos que se diseñara, porque ofrecen muchas más ventajas sobre los PICs 16FXXX y mas por la cantidad de memoria que se va a ocupar dentro del circuito que maneja el GLCD.

**Características de los Microcontroladores 18F452:**

- Código compatible con los PICs 16Fxxx y 17Fxx.
- Reloj de cristal de 4 a 10 MHz.
- Instrucciones de 16 bits con bus de datos de 8 bits.

- Tres pines para manejo de interrupciones externas.
- Timer 1 de 16bits, timer 2 de 8bits, timer 3 de 16 bits.
- Módulos serial con soporte para RS-485 y RS-232.
- Memoria Flash de 32k bytes.
- Almacena 16mil instrucciones simples.
- Constan de 3 registros para su operación:
  - TRIS: Controla la dirección de funcionamiento del puerto.
  - PORT: Lee los niveles de entrada en el puerto.
  - LAT: Es el LATCH de salida del puerto.
- Conversión A/D posee una resolución de 10 bits controlados por los registros ADCON1 y ADCOM0.

**Nota:** El Datasheet del microcontrolador 18F452 se encuentra en el Anexo 12.



Figura 14: Fotografía PIC 18F452

### 3.3.7. SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA SAI - UPS

Siempre que se cuenta con sistemas eléctricos se debe tomar en cuenta las fallas eléctricas que se puedan ocasionar y se debe prever un sistema que respalde el funcionamiento del sistema por cualquier eventualidad que ocurra, y más aún si se trata de un sistema de seguridad.



## Tipos de UPS

- Standby
- Línea Interactiva
- Standby-Ferro
- Doble Conversión On-Line
- Delta Conversion On-Line

En la siguiente tabla, se mostrará una breve comparación entre los tipos de UPS.

	Costo	Eficiencia	Tasa de energía (kVA)	Convertidor siempre operando
Standby	Bajo	Muy Alto	0 – 0.5	No
Línea Interactiva	Medio	Muy Alto	0.5 - 5	Depende del diseño
Standby Ferro	Alto	Bajo - Medio	3 - 15	No
Doble Conversión On-Line	Medio	Bajo – Medio	5 - 5000	Si
Conversión Delta On-Line	Medio	alto	5 - 5000	Si

Tabla 7: Comparación de tipos de UPS

Como método más adecuado se ha optado por el uso de un UPS (Uninterrupted Power System o Sistema de Potencia Ininterrumpida) de tipo Standby, que estará conectado al equipo servidor y dará un tiempo aproximado de 15 minutos para tomar medidas con relación a la seguridad del lugar donde esté implementado el sistema. Por otro lado como sistema de seguridad contra el corte de energía para los demás circuitos se hará uso de baterías de 9 voltios a excepción del circuito que contendrá el GLCD.

Al requerir un mayor grado de alimentación de corriente el módulo que contiene el GLCD, utilizará una batería de 6 voltios y 3 Amperios que también sirve de respaldo en caso de fallas eléctricas y hace posible manejar el control Táctil de forma inalámbrica dentro del hogar.



Figura 15: Fotografía Batería 6V – 1.3A



Figura 16: Fotografía Batería 9V

### 3.4. CONEXIÓN A INTERNET

Se hace indispensable contar con una conexión a internet de forma permanente en el servidor web que utilizará el sistema, debido a que para el envío de correos electrónicos, vigilancia por medio de cámara web y manejo de la aplicación web, se requiere una conexión de buena disponibilidad.

#### Anchos de Banda ADSL:

- **128 x 64 Mbps.**  
Envío de correos, chat, navegación limitada
- **256 x 128 Mbps.**  
Envío de correos, chat, navegación más rápida, descarga de música
- **512 x 128 Mbps.**  
Envío de correos, chat, navegación más rápida, descarga de música y videos, video conferencia

- **1024 x 512 Mbps.**

Envío de correos, chat, navegación mucho más rápida, descarga de música y videos en pocos minutos, video conferencia, descarga de programas de alto peso etc.

Analizando los servicios a los que se tiene acceso de manera más rápida y dependiendo de las velocidades anteriormente expuestas, se concluye que la más adecuada para el óptimo desempeño del sistema es la de 1024 x 512 Mbps, debido a que el servidor Web estará permanentemente conectado a la Web y el envío de imágenes por medio de la cámara IP es continuo, se requiere un tipo de conexión con una tasa de transferencia mucho más alta se subida.

También dependerá de la conexión con la cual se conecten remotamente los clientes al servidor para tener un servicio claro y rápido, se recomienda trabajar con la misma velocidad del servidor o en el peor de los casos adoptar una velocidad de 512 x 128 Mbps.

### **3.5. REQUERIMIENTOS BÁSICOS DEL SERVIDOR**

#### **3.5.1. REQUERIMIENTOS MÍNIMOS**

- Microsoft Windows XP Professional SP2.
- Procesador: 1 GHz o superior.
- Memoria RAM: 512 GB.
- Espacio en disco: 2 GB de espacio libre en disco.

#### **3.5.2. REQUERIMIENTOS RECOMENDADOS**

- Microsoft Windows XP Professional SP2.
- Procesador: 2.6 GHz Intel Pentium IV o equivalente.
- Memoria RAM: 2 GB.
- Espacio en disco: 4 GB de espacio libre en disco.
- Tarjeta gráfica: 256 MB o superior.

## **CAPÍTULO 4**

### **DISEÑO DEL PROYECTO**

Este capítulo hace referencia al diseño de la solución bajo criterios establecidos, diseño de hardware y de software de cada uno de los componentes del sistema domótico.

#### **4.1. CRITERIOS PARA EL DISEÑO DE LA SOLUCIÓN**

En la aparición de nuevas tecnologías o servicios, el grado de importancia de la parte técnica es muy alto y se tiende a complicar su uso, por la incorporación de cientos de instrucciones, funciones, programaciones, etc. En el caso de servicios o sistemas orientados a usuarios finales, esta situación se complica, porque el sistema puede ser técnicamente aceptable pero en la práctica el usuario podría encontrarse con eventos que le producirán confusión, desconcierto y finalmente el rechazo.

Estará basado en dos tipos de criterios para el diseño de un sistema domótico, criterios técnicos y del usuario.

##### **4.1.1. CRITERIOS TÉCNICOS**

- Topología de red.
- Tipo de arquitectura del sistema domótico.
- Medios de transmisión.
- Seguridad.
- Tipos de Protocolo.
- Dificultad de construcción.
- Tolerancia a fallos.

##### **4.1.2. CRITERIOS DEL USUARIO**

- Simplicidad de uso.
- Fácil ampliación (Escalabilidad).
- Análisis costo/beneficio aceptable.

- Confiabilidad.
- Disponibilidad.
- Variedad de funciones y elementos de control que satisfagan necesidades de seguridad y confort.

### **Medios de transmisión**

En el estudio de análisis de requerimientos se establece usar:

- Zigbee para la comunicación entre módulos.
- Ethernet para la comunicación con la cámara, Router WiFi, modem ADSL, servidor de BDD y servidor web.
- Bluetooth para la comunicación entre el servidor y un dispositivo móvil mediante una aplicación bluetooth.
- GSM para la comunicación entre el servidor y un teléfono celular que envíe y reciba SMS.

### **Topología**

La topología de red que se implementa en este sistema domótico es malla o Mesh, brinda un alto grado de disponibilidad ante posibles fallas en los nodos y proporciona un alcance de distancia superior que cualquier otra topología no posee.

### **Tipo de arquitectura**

El tipo de arquitectura utilizado en la domótica se refiere al modo en que se van a ubicar los diferentes elementos de control del sistema, es decir, este criterio se refiere a donde reside la inteligencia del sistema domótico.

Se adoptó la arquitectura mixta, que consta de varios 4 módulos que interpretan a un determinado grupo de sensores y actuadores, y un quinto módulo que interpreta toda la información ver Figura 17.

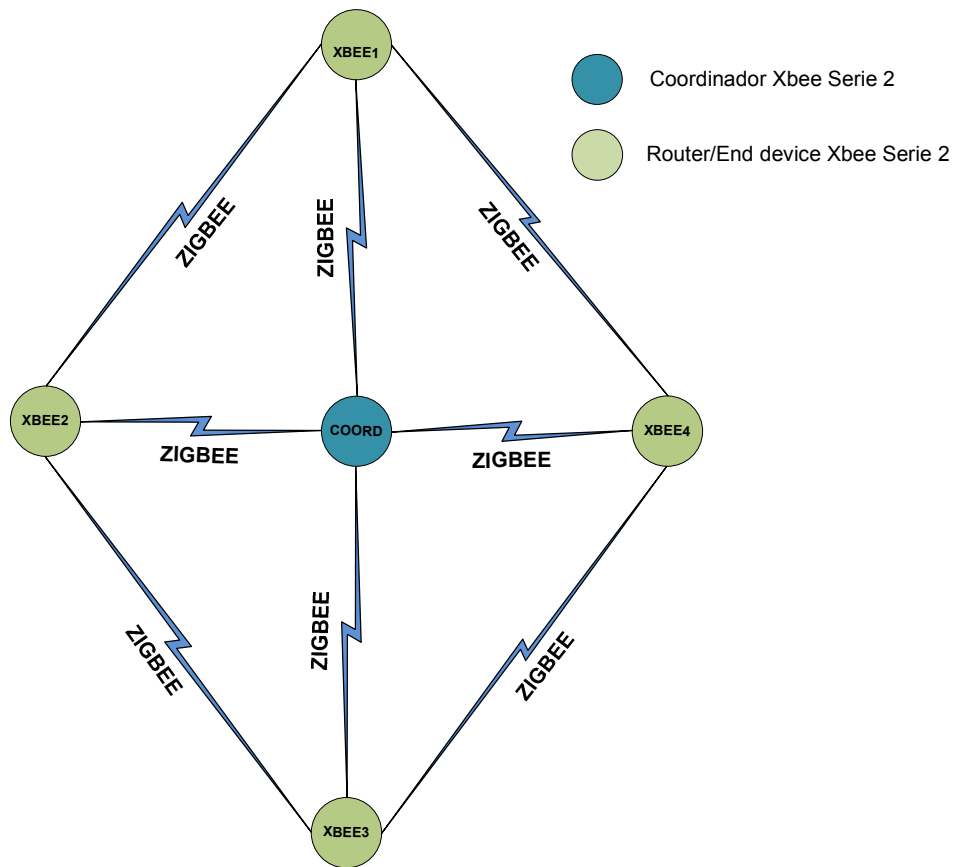


Figura 17: Distribución de los Xbee

## 4.2. DISEÑO DE LA SOLUCIÓN

El siguiente gráfico, se detalla con claridad la topología, tipo de arquitectura y medios de transmisión usados en este diseño, también están detallados los sensores y actuadores que tendrá conectado cada módulo Xbee.

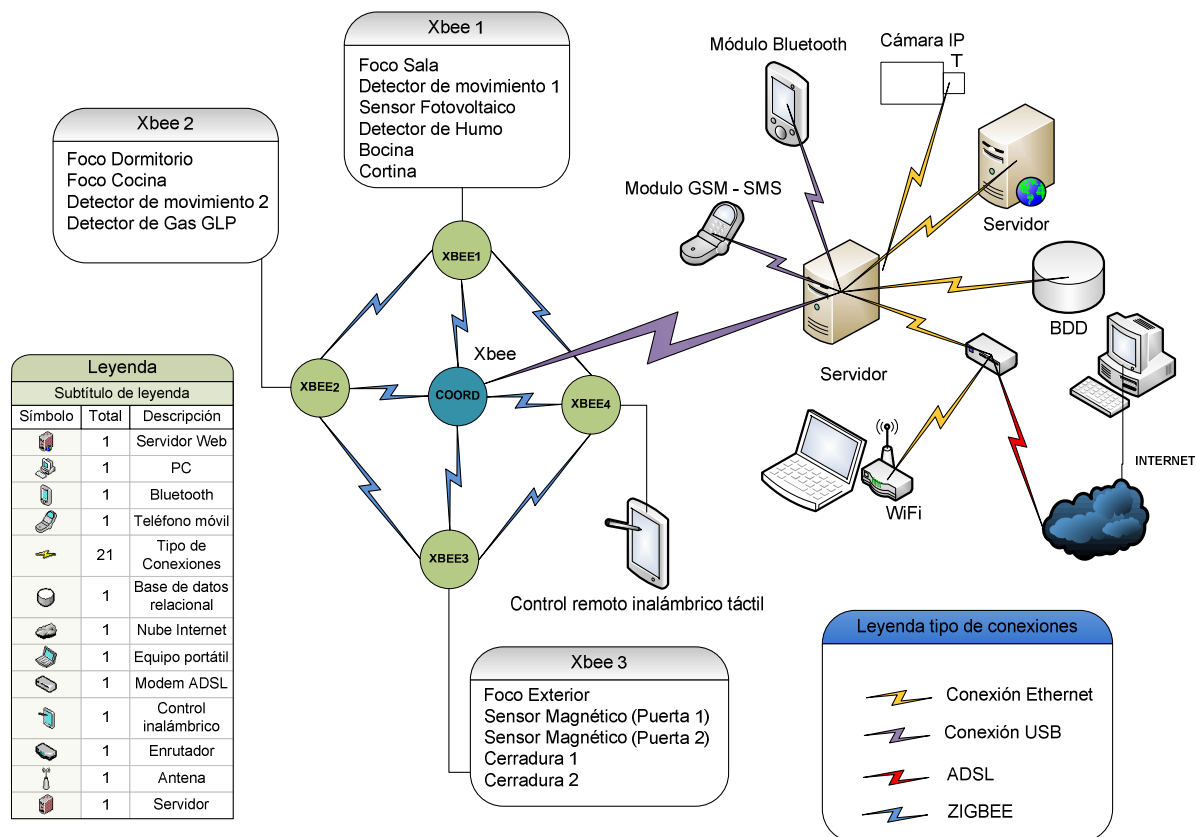


Figura 18: Diseño de la solución

**Nota:** Los planos de distribución se observa en el Anexo 13.

El análisis costo beneficio es importante para garantizar el éxito en el desarrollo del proyecto, se estudia que tan viable son los costos, construcción e implementación versus los beneficios que el sistema ofrece.

**Nota:** El análisis costo beneficio se observa en el Anexo 18.

## 4.2.1. ZIGBEE

### 4.2.1.1. Módulos de comunicación

Se divide el sistema en 5 módulos uno para cada nodo Zigbee (Xbee series 2), se definen los dispositivos finales conectados a cada módulo y sus respectivos nombres:

- **Módulo Coordinador:** Conectado directamente al servidor por medio de una interfaz USB con un FT232RL que emula un puerto COM, de esta manera se puede establecer la comunicación serial entre el servidor y el coordinador.

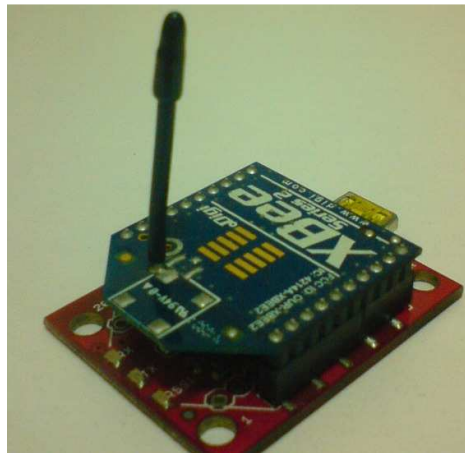


Figura 19: Módulo Coordinador

- **Módulo Xbee1 :** Un módulo Zigbee (Xbee series 2) conectado a un PIC 18F452 y este a su vez conectado a:
  - Foco Sala
  - Detector de movimiento 1
  - Sensor Fotovoltaico
  - Detector de Humo
  - Bocina
  - Cortina



- **Módulo Xbee2** : Un módulo Zigbee (Xbee series 2) conectado a un PIC 18F452 y este a su vez conectado a:
  - Foco Dormitorio
  - Foco Cocina
  - Detector de movimiento 2
  - Detector de Gas GLP
  
- **Módulo Xbee3** : Un módulo Zigbee (Xbee series 2) conectado a un PIC 18F452 y este a su vez conectado a :
  - Foco Exterior
  - Sensor Magnético (Puerta 1)
  - Sensor Magnético (Puerta 2)
  - Cerradura 1
  - Cerradura 2
  
- **Módulo GLCD (Xbee4)** : Un módulo Zigbee (Xbee series 2) conectado a un PIC 18F452 y este a su vez conectado a:
  - GLCD 240x128 Touch

#### 4.2.1.2. Nomenclatura de identificación

Para que el sistema identifique de manera sencilla cada uno de los módulos y dispositivos finales es necesario diseñar una nomenclatura en la que se especifique el número de módulo, tipo y número de dispositivo final así como el estado en que se encuentra, sea encendido/apagado activado/desactivado.

De esta manera, es más fácil identificar la gran cantidad de dispositivos finales que tiene este sistema.

#### **Nomenclatura para los módulos de comunicaciones**

<b>Módulos Xbee</b>	
Xbee 1	1
Xbee 2	2
Xbee 3	3
GLCD (Xbee 4)	4
Coordinador	5

**Tabla 8: Nomenclatura para los módulos de comunicaciones**

#### **Nomenclatura para el tipo de dispositivo final**

<b>Tipo de dispositivo final</b>	
Luces	1
Detector de movimiento	2
Detector de humo	3
Detector de gas	4
Sensor magnético	5
Cerradura eléctrica	6
Sensor fotovoltaico	7
Bocina	8
Cortinas eléctricas	9

**Tabla 9: Nomenclatura para el tipo de dispositivo final**

## Nomenclatura para los dispositivos finales

Luces	
Dormitorio	1
Sala	2
Cocina	3
Exterior	4

Tabla 10: Nomenclatura dispositivo final luces

Detectores de movimiento	
Detector de movimiento 1	1
Detector de movimiento 2	2

Tabla 11: Nomenclatura dispositivo final detectores de movimiento

Detector de humo	
Detector de humo	1

Tabla 12: Nomenclatura dispositivo final detector de humo

Cerraduras eléctricas		
Cerradura 1	Puerta 1	70
Cerradura 2	Puerta 2	71

Tabla 13: Nomenclatura dispositivo final Cerraduras eléctricas

Sensores magnéticos		
Sensor 1	Puerta 1	1
Sensor 2	Puerta 2	2

Tabla 14: Nomenclatura dispositivo final Sensores magnéticos

Detector de gas		
Detector de gas	1	

Tabla 15: Nomenclatura dispositivo final detector de GAS GLP

Sensor fotovoltaico		
Sensor fotovoltaico	Exterior	2

Tabla 16: Nomenclatura dispositivo final sensor fotovoltaico

Bocina		
Bocina	1	72

Tabla 17: Nomenclatura dispositivo final Bocina

### Nomenclatura para estado del dispositivo final

Estado	
Encendido / Activado	1
Apagado / Desactivado	0

Tabla 18: Nomenclatura estado del dispositivo final

En las siguientes figuras, se muestra la nomenclatura de los códigos de comunicación que van a identificar a cada uno de los dispositivos finales junto con su estado.

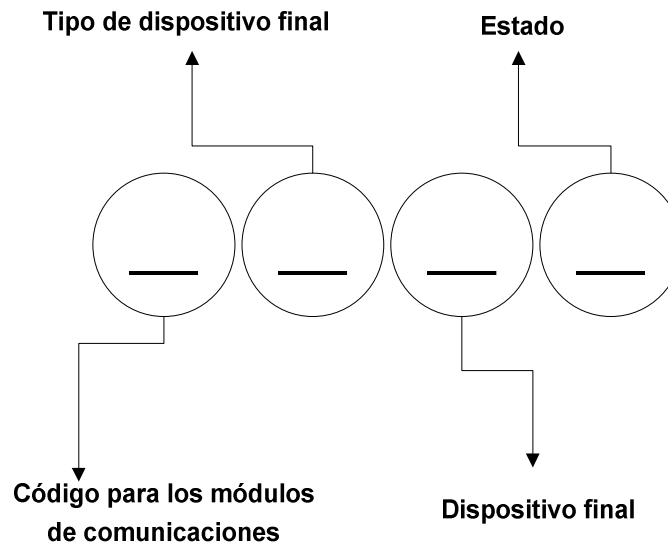


Figura 20: Nomenclatura

#### 4.2.1.3. Códigos de comunicación

Los códigos de comunicación se usan principalmente para reducir las tramas de datos enviadas entre el cliente y el servidor, de la misma manera el uso de códigos en la comunicación provee de un nivel de seguridad adicional.

##### 4.2.1.3.1. Enviados hacia el servidor

Los códigos enviados por los módulos de comunicación hacia el servidor o módulo coordinador cuando un evento sucede son los siguientes:

Componiendo la nomenclatura se tiene las siguientes combinaciones para cada dispositivo final indicando si está Encendido/Activado o Apagado/Desactivado.

#### Xbee1

Foco sala	1	1	2	0/1
Detector de movimiento 1	1	2	1	0/1
Sensor fotovoltaico	1	7	1	0/1
Detector de humo	1	3	1	0/1
Bocina	1	8	1	0/1
Cortinas eléctricas	1	9	1	0/1
Falla de energía	FallaX1			
Energía restablecida	CorrectoX1			

Tabla 19: Códigos de comunicación Xbee 1 enviados hacia el servidor

#### Xbee2

Foco dormitorio	2	1	1	0/1
Detector de movimiento 2	2	2	2	0/1
Foco cocina	2	1	3	0/1
Detector de gas	2	4	1	0/1
Falla de energía	FallaX2			
Energía restablecida	CorrectoX2			

Tabla 20: Códigos de comunicación Xbee 2 enviados hacia el servidor

### Xbee3

Foco del exterior	3	1	4	0/1
Puerta 1 sensor magnético 1	3	5	1	0/1
Puerta 2 sensor magnético 2	3	5	2	0/1
Cerradura 1	3	6	1	0/1
Cerradura 2	3	6	2	0/1
Falla de energía	FallaX3			
Energía restablecida	CorrectoX3			

Tabla 21: Códigos de comunicación Xbee 3 enviados hacia el servidor

### GLCD

Para el módulo GLCD se necesitó códigos de comunicación distintos, porque en este módulo se valida el ingreso del tipo de usuario y nombre de usuario, también peticiones de encendido o apagado de los diferentes dispositivos finales que van hasta el servidor para ahí ser interpretados.

Control de usuario		
Solicitud de emergencia	"EMERGENCIA"	
Selecciono tipo de usuario	Administrador	"1"
	Invitado	"2"
Envío clave 4 dígitos	Clave	

Tabla 22: Códigos de comunicación Control de usuario, enviados hacia el servidor

Gestión de la iluminación	
Apagar/Encender SALA	"SALA"
Apagar/Encender EXTERIOR	"EXT"
Apagar/Encender COCINA	"COCI"
Apagar/Encender DORMITORIO	"DORMI1"

Activar/Desactivar Modo Ahorro de Energía	"MAE"
Activar/Desactivar Modo Simulador de presencia	"MP"

Tabla 23: Códigos de comunicación Gestión de la iluminación enviados hacia el servidor

Monitoreo de sensores	
Petición de estado sensor magnético 1	"puerta1"
Petición de estado sensor magnético 2	"puerta2"
Petición de estado detector de movimiento 1	"mov1"
Petición de estado detector de movimiento 2	"mov2"
Petición de estado detector de humo	"humo"
Petición de estado detector de gas GLP	"gas"
Petición de estado detector de sensor fotovoltaico	"luz"

Tabla 24: Códigos de comunicación Monitoreo de sensores enviados hacia el servidor

Control de cerraduras	
Activar cerradura 1	"cerradura1"
Activar cerradura 2	"cerradura2"
Activar/Desactivar bocina	"bocina"

Tabla 25: Códigos de comunicación Control de Cerraduras enviados hacia el servidor



#### 4.2.1.3.2. Enviados desde el servidor

##### Xbee1

	Letra	Decimal
Apagar/Encender Foco sala	'A'	65
Apagar/Encender Bocina	'H'	72
Abrir cortina	'I'	73
Cerrar cortina	'J'	74
Petición de estado	'x'	120

Tabla 26: Códigos de comunicación Xbee 1 enviados desde el servidor

##### Xbee2

	Letra	Decimal
Apagar/Encender Foco dormitorio	'D'	68
Apagar/Encender Foco cocina	'C'	67
Petición de estado	'y'	121

Tabla 27: Códigos de comunicación Xbee 2 enviados desde el servidor

##### Xbee3

	Letra	Decimal
Apagar/Encender Foco del exterior	'B'	66
Activar cerradura 1	'F'	70
Activar cerradura 2	'G'	71
Petición de estado	'z'	122

Tabla 28: Códigos de comunicación Xbee 3 enviados desde el servidor

## GLCD

	Letra	Decimal
Contraseña correcta	's'	115
Contraseña incorrecta	'n'	110
Encendido / Activado	'p'	112
Apagado / Desactivado	'o'	111

Tabla 29: Códigos de comunicación GLCD enviados desde el servidor

Para mostrar el estado Encendido/Activado de cualquier dispositivo final el servidor responde con la letra 'p' (Decimal 112)

Para mostrar el estado Apagado/Desactivado de cualquier dispositivo final el servidor responde con la letra 'o' (Decimal 111)

### 4.2.2. BLUETOOTH

La aplicación Bluetooth va a estar diseñada solo para uso de tipo invitado es decir, el tipo de usuario por defecto será invitado, solo permite gestionar la iluminación y monitorear los sensores de movimiento y magnéticos ubicados en las puertas.

#### 4.2.2.1. Interface de comunicación

La interface de comunicación bluetooth que se usa para este proyecto es una D-link DBT-122 USB y manejando su puerto COM se puede establecer conexión con otro dispositivo Bluetooth que acepte la conexión.



Figura 21: Fotografía Bluetooth Dlink USB

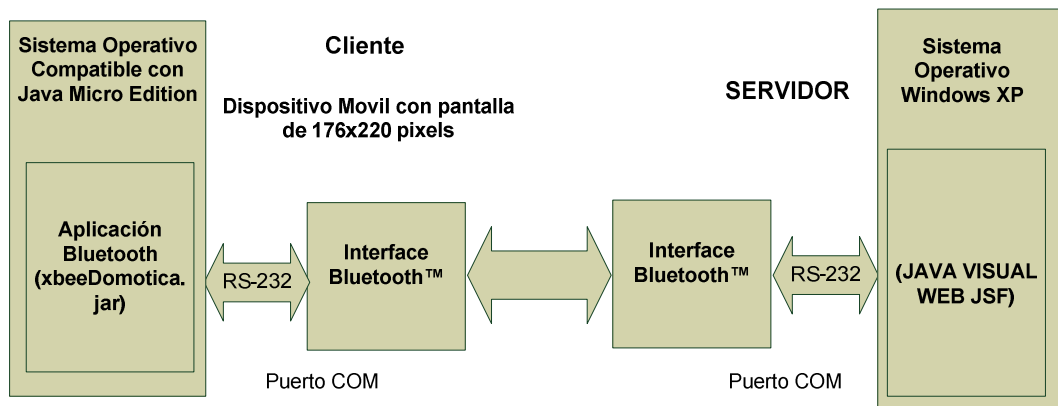


Figura 22: Comunicación Bluetooth

El cliente Bluetooth realiza varias peticiones previas al servidor una de ellas es buscar puertos disponibles, una vez encontrado un puerto el cliente pregunta al servidor si puede hacer uso de este puerto y el servidor debe aceptarlo o denegarlo.

Para realizar esto primero se debe emparejar el dispositivo móvil con el servidor proporcionando una clave que será digitada una sola vez tanto en el cliente como en el servidor.

Una vez que se tenga el cliente emparejado al servidor por medio de un Hyperterminal, se prueba que están llegando los datos al puerto COM del servidor.

#### 4.2.2.2. Códigos de comunicación

Los códigos de comunicación se usan principalmente para reducir las tramas de datos enviadas entre el cliente y el servidor aumentando la velocidad de transmisión, de la misma manera el uso de códigos en la comunicación provee de un nivel de seguridad adicional.

##### 4.2.2.2.1. Enviados hacia el servidor

Los códigos enviados por el cliente hacia el servidor son para la petición de control de usuario, gestionar la iluminación y pedir el estado de los sensores magnéticos y de movimiento.

Control de usuario	
Solicitud de emergencia	"EMERGENCIA"
Ingreso usuario	Usuario
Ingreso contraseña	Contraseña
Al dar clic en Ingresar se envía al servidor	Usuario. Contraseña

Tabla 30: Códigos de comunicación Control de usuario Bluetooth enviados hacia el servidor

Gestión de la iluminación	
Apagar/Encender SALA	'A'
Apagar/Encender EXTERIOR	'B'
Apagar/Encender COCINA	'C'
Apagar/Encender DORMITORIO	'D'

Tabla 31: Códigos de comunicación Gestión de iluminación Bluetooth enviados hacia el servidor

Monitoreo	
Petición de estado de los sensores de mov. y magnéticos	"ESTADO"

Tabla 32: Códigos de comunicación Monitoreo Bluetooth enviados hacia el servidor

#### 4.2.2.2.2. Enviados desde el servidor

Los códigos enviados por el servidor hacia el cliente son para responder la petición de control de usuario y mostrar el estado de los sensores magnéticos y de movimiento.

#### Control de usuario

	Letra	Decimal
Contraseña correcta	's'	115
Contraseña incorrecta	'n'	110

Tabla 33: Códigos de comunicación Control de usuario Bluetooth enviados desde el servidor

#### Monitoreo

	Letra	Decimal
Detector de mov1 Activado	'1'	49
Detector de mov1 Desactivado	'2'	50
Detector de mov2 Activado	'3'	51
Detector de mov2 Desactivado	'4'	52
Sensor magnético 1 Activado	'5'	53
Sensor magnético 1 Desactivado	'6'	54
Sensor magnético 2 Activado	'7'	55
Sensor magnético 2 Desactivado	'8'	56

Tabla 34: Códigos de comunicación Monitoreo Bluetooth enviados desde el servidor

### 4.2.3. SMS

La comunicación SMS se realiza con un modem GSM que reconozca comandos AT, para ello es recomendable usar el celular Nokia 3220b que se comporta como un modem conectado al computador y reconoce fácilmente los comandos AT suministrados a su puerto COM por medio de un Hyperterminal, es económicamente muy recomendable.

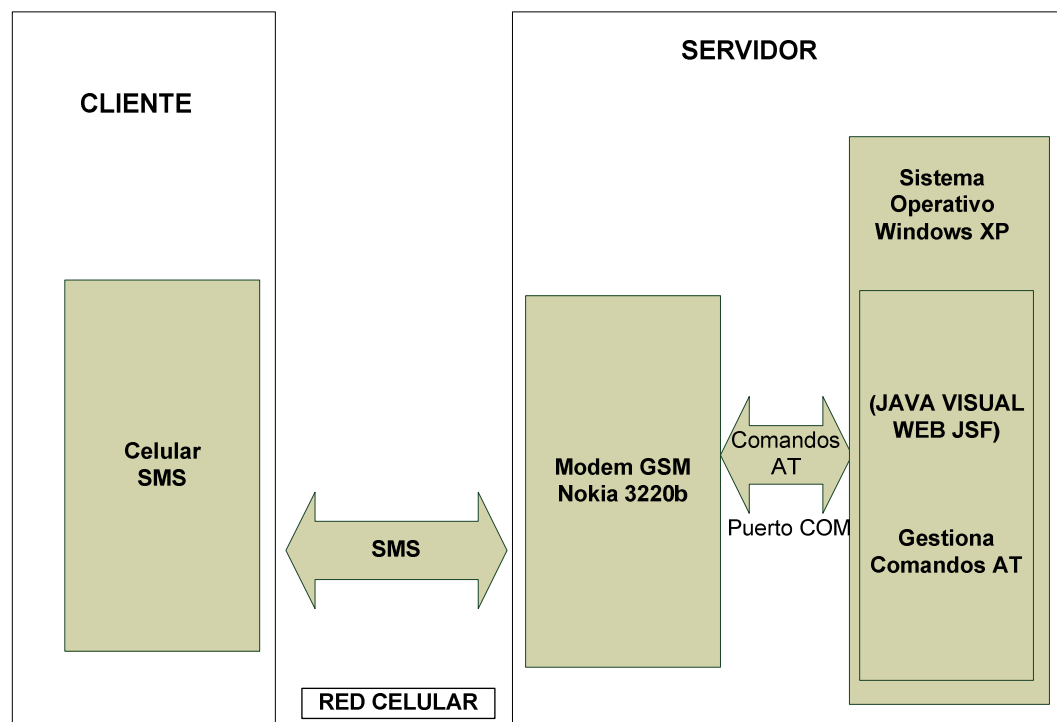


Figura 23: Comunicación SMS

El servidor gestiona al modem GSM mediante los comandos AT ya sea para enviar o leer un mensaje de texto SMS, de esta manera es posible establecer comunicación con cualquier dispositivo celular que tenga la posibilidad de lectura y escritura mensajes de texto SMS.

Para que la comunicación entre el cliente y el servidor se lleve a cabo satisfactoriamente, el número de celular del cliente debe estar registrado en uno de los usuarios almacenados en la base de datos del servidor.

#### 4.2.3.1. Comandos de comunicación

Con el fin de facilitar la comunicación entre el cliente y el servidor se han establecido una serie de comandos de comunicación capaces de controlar la iluminación, monitoreo y cerraduras.

##### 4.2.3.1.1. Enviados hacia el servidor

Los comandos no son susceptibles a las mayúsculas o minúsculas por lo tanto no las toma en cuenta.

Gestión de la iluminación	
Encender foco sala	"prende sala"
Apagar foco sala	"apaga sala"
Encender foco exterior	"prende exterior"
Apagar foco exterior	"apaga exterior"
Encender foco cocina	"prende cocina"
Apagar foco cocina	"apaga cocina"
Encender foco dormitorio	"prende dormitorio"
Apagar foco dormitorio	"apaga dormitorio"
Desactivar toda la iluminación (Apagar todo)	"apaga todo"
Activar toda la iluminación (Enciende todo)	"prende todo"

Tabla 35: Códigos de comunicación Gestión de iluminación SMS enviados hacia el servidor

Cerraduras	
Abrir cerradura 1	"cerradura1 " + cod_usuario
Abrir cerradura 2	"cerradura2 " + cod_usuario

Tabla 36: Códigos de comunicación Cerraduras SMS enviados hacia el servidor

Monitorear Estados	
Petición de estado de la iluminación	"estado ilu"
Petición de estado detectores de movimiento	"estado mov"
Petición de estado sensores magnéticos	"estado mag"
Petición de estado detectores de humo, gas y fotovoltaico	"estado det"

Tabla 37: Códigos de comunicación Monitorear estados SMS enviados hacia el servidor

El comando ayuda permite recibir la lista de los comandos existentes para gestionar el sistema domótico.

Ayuda	
Petición de ayuda para los comandos de iluminación y cerraduras	"ayuda"
Petición de ayuda para los comandos de monitoreo	"ayuda moni"

Tabla 38: Códigos de comunicación Ayuda SMS enviados hacia el servidor

#### 4.2.3.1.2. Enviados desde el servidor

Monitoreo ofrecido por el servidor para cambios en los estados de los diferentes dispositivos finales existentes.

Monitoreo de la iluminación	
Foco sala encendido	"Foco sala encendido"
Foco sala apagado	" Foco sala apagado "
Foco exterior encendido	" Foco exterior encendido "
Foco exterior apagado	" Foco exterior apagado "
Foco cocina encendido	" Foco cocina encendido "
Foco cocina encendido	" Foco cocina encendido "



Foco dormitorio encendido	" Foco dormitorio encendido "
Foco dormitorio apagado	" Foco dormitorio apagado "
Desactivar toda la iluminación (Apagar todo)	"apaga todo"
Activar toda la iluminación (Enciende todo)	"prende todo"
Petición de estado de la iluminación	"estado"

Tabla 39: Códigos de comunicación Monitoreo de iluminación SMS enviados desde el servidor

Monitoreo de sensores	
Detector de mov1 Activado	" Detector de mov1 Activado "
Detector de mov1 Desactivado	" Detector de mov1 Desactivado "
Detector de mov2 Activado	" Detector de mov2 Activado "
Detector de mov2 Desactivado	" Detector de mov2 Desactivado "
Sensor magnético 1 Activado	" Sensor magnético 1 Activado "
Sensor magnético 1 Desactivado	" Sensor magnético 1 Desactivado "
Sensor magnético 2 Activado	" Sensor magnético 2 Activado "
Sensor magnético 2 Desactivado	" Sensor magnético 2 Desactivado "
Detector de humo Activado	" Detector de humo Activado "
Detector de gas Activado	" Detector de gas Activado "

Tabla 40: Códigos de comunicación Monitoreo de sensores SMS enviados desde el servidor

Ayuda	
Petición de ayuda	" Comandos básicos : prende [ubic] / apaga [ubic]; ubic = sala,cocina,exterior,dormitorio ; prende todo / apaga todo ; cerradura1 cod_usuario "
Petición de ayuda monitoreo	"Comandos monitoreo : estado ilu ; estado mov ; estado mag ; estado det"

Tabla 41: Códigos de comunicación Ayuda SMS enviados desde el servidor

#### 4.2.4. CORREO ELECTRÓNICO

Para la configuración de correo electrónico no es necesario levantar un servidor de correo propio, se puede usar GMAIL ofrece muchas ventajas y es más sencillo que levantar uno desde cero.

Lo principal es iniciar la cuenta de correo de GMAIL desde el servidor local usando dos librerías básicas para este propósito mail.jar y activation.jar, se configuran los parámetros de la cuenta de GMAIL tales como dirección de correo, contraseña, etc. Una vez que este iniciada se podrá enviar correos de notificación a cualquier dirección cuando se requiera.

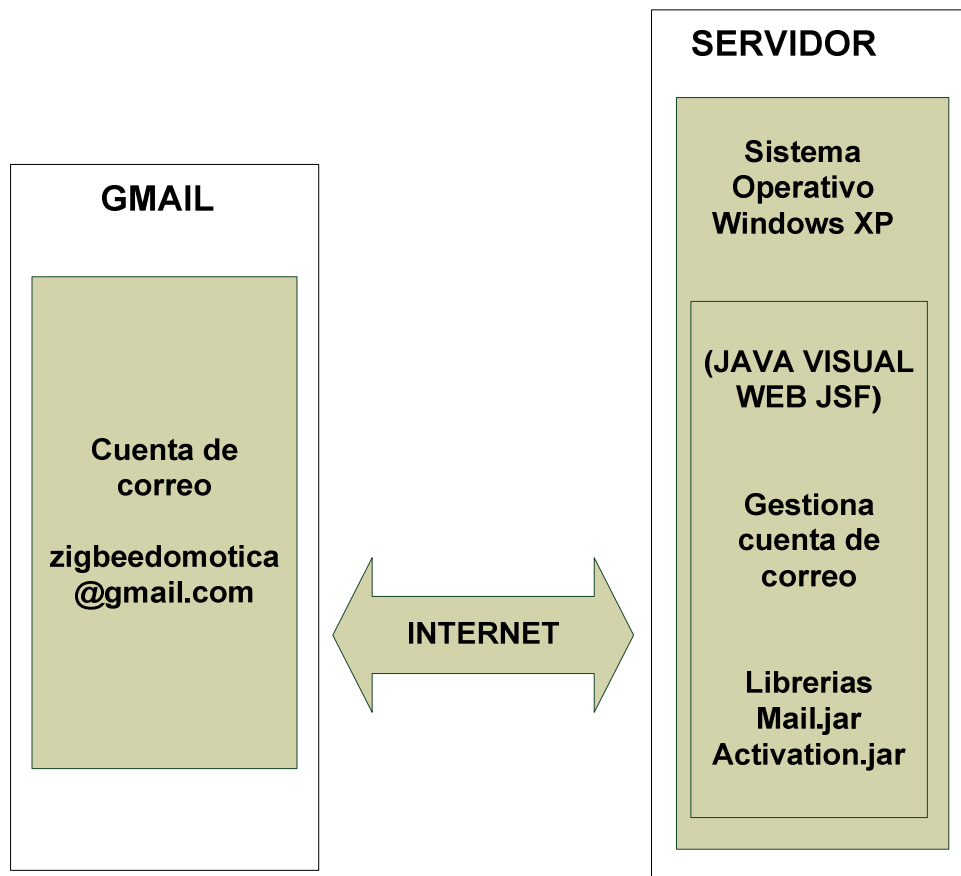


Figura 24: Comunicación con el servidor de correo

#### **4.2.5. CÁMARA IP**

La cámara IP se muestra en la aplicación JSF mediante un applet embebido en JSP controlando el zoom en 4 niveles. Es muy sencillo mostrar la imagen que envía la cámara, solo es necesario identificarla dentro de la red con una dirección IP estática.

#### **4.2.6. SERVIDOR PRINCIPAL**

El servidor principal corre bajo el sistema operativo Windows XP, la aplicación web realizada en java visual web JSF está alojada en el servidor web Tomcat usando el puerto 8080, la base de datos relacional está alojada en el servidor de base de datos MySQL usando el puerto 3306, ambos servidores están compartiendo el mismo computador pero usando puertos distintos.

El servidor principal ofrece 4 servicios básicos:

##### **Monitoreo**

El servicio de monitoreo permite al usuario administrador o invitado saber en todo momento y lugar el estado de todos los dispositivos finales del sistema.

Visualizar mediante una Cámara IP imágenes en tiempo real del hogar.

##### **Control**

Este servicio permite controlar los diferentes dispositivos finales actuadores (iluminación, Cerraduras, Cortina, Bocina).

Dentro de control existen los denominados “MODOS” en este sistema existen 3 modos principales que son.

**Modo Simulador de Presencia (MSP):** Este modo como su nombre lo indica consiste en simular presencia en el hogar, en tiempos aleatorios el sistema automáticamente gestiona la iluminación del hogar enciende o apaga focos y abre o cierra la cortina.

**Modo Ahorro de Energía (MAE):** Este modo como su nombre lo indica consiste en ahorrar energía, se gestiona por medio de dos sensores principales, los detectores de movimiento y el sensor fotovoltáico.

Cuando se activa un detector de movimiento el foco que esté relacionado a él se encenderá, si no detecta movimiento en la habitación por un determinado tiempo el foco se apagará.

El sensor fotovoltáico detecta el día y la noche por lo tanto, el foco exterior está relacionado directamente, cuando anochece el foco exterior se enciende y cuando amanece se apaga de esta manera se consigue ahorrar cantidades importantes de energía, que eran desperdiciadas innecesariamente.

**Modo Alarma Sonora (MAS):** Este modo como su nombre lo indica consiste en activar una alarma sonora (Bocina) cada vez que se active uno de estos sensores:

Detectores de movimiento

Detector de Humo

Detector de GAS

Sensores Magnéticos

El usuario administrador puede seleccionar uno o varios de estos sensores, para que cuando ocurra un evento active la alarma sonora.

## Notificaciones

Las notificaciones que el sistema puede enviar al usuario pueden ser de 2 formas, por medio de:

- SMS
- Correo electrónico

## Gestión

La gestión de la base de datos se basa en consultar, modificar, insertar y eliminar los datos de las tablas.

Los módulos de comunicaciones GSM, Bluetooth y Zigbee, están conectados al servidor mediante USB usando transmisión serial por medio de los puertos COM 11, COM 10 y COM 17. Como se indica en la Figura 25.

La cámara IP está conectada al servidor por medio de Ethernet usando cable UTP teniendo una dirección ip propia que la identifica en la red.

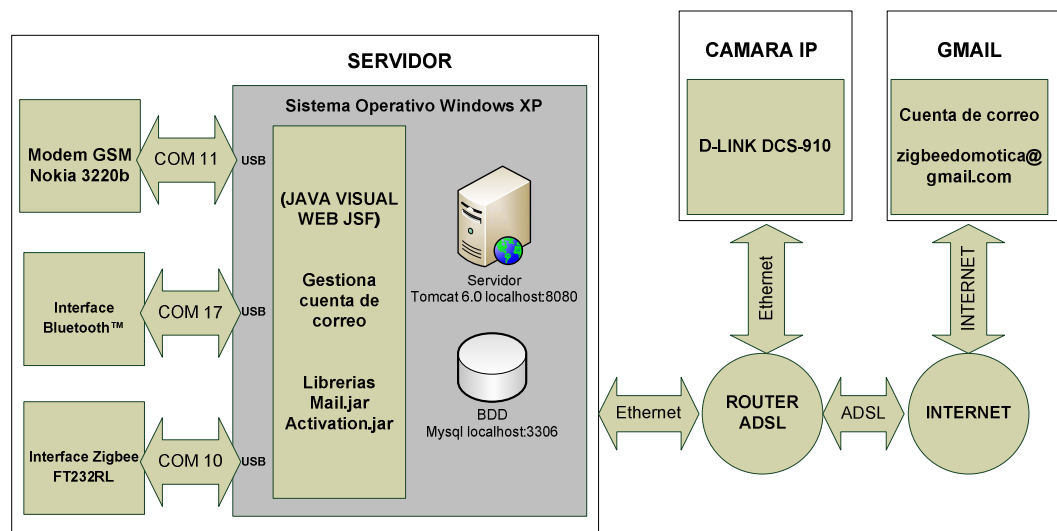


Figura 25: Diagrama de bloques servidor principal

### **4.3. DISEÑO DE HARDWARE**

#### **4.3.1. MÓDULO XBEE1**

Para el diseño del módulo Xbee1 se usa el PIC 18F452 de gama alta que tiene gran capacidad de memoria en relación a los de gama baja.

Un módulo Xbee series 2 conectado a los puertos seriales del microcontrolador RC6 TX y RC7 RX para establecer comunicación con el coordinador.

Dos Leds de estado conectados al módulo Xbee para verificar si esta encendido y escuchando.

## Diagrama de Bloques “ XBEE1 ”

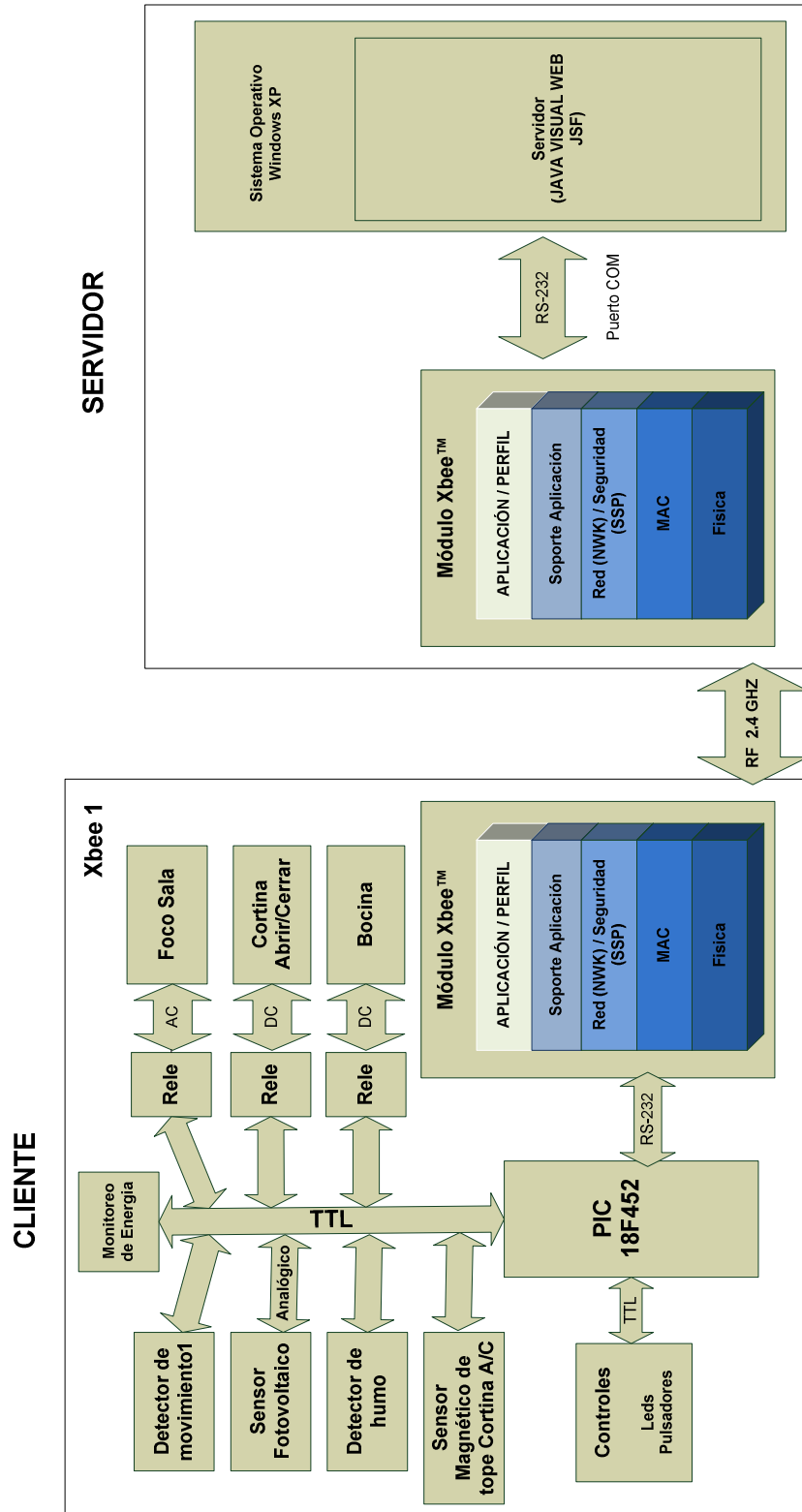


Figura 26: Diagrama de Bloques Módulo Xbee 1

#### **4.3.1.1. Actuadores**

##### **Foco Sala**

Desde una salida del microcontrolador se conecta un transistor que activa un relé de 12v DC y este a su vez activa el Foco de la sala de 110v AC.

Usando otra salida del microcontrolador se activa un led de estado que indica si el foco esta encendido o apagado.

##### **Bocina**

Desde una salida del microcontrolador se conecta un transistor que activa un relé de 12v DC y este a su vez activa la bocina de 12v conectada a otra fuente debido a su alto consumo 1A.

Usando otra salida del microcontrolador se activa un led de estado que indica si la bocina esta activada o desactivada.

##### **Cortina**

El diseño del hardware que controlara la cortina es un poco complejo porque se debe manejar apertura y cierre, también usar dos sensores que muestren el fin de ciclo, es decir, que informen cuando la cortina está totalmente abierta o totalmente cerrada con el fin de detener el motor.

Dos salidas del microcontrolador una para apertura y otra para cierre de la cortina, conectadas cada una a un transistor y este a un relé de 12v.

Dos entradas del microcontrolador conectadas a dos sensores magnéticos de fin de ciclo.

Usando dos salidas mas del microcontrolador activa o desactiva dos LED's de estado para saber si la cortina está abierta o cerrada.



#### **4.3.1.2. Sensores**

##### **Detector de movimiento**

Envía un pulso bajo cuando esta activado y un pulso alto cuando esta desactivado, esta conectado a una entrada digital del microcontrolador.

##### **Detector de humo**

El detector de humo envía pulsos TTL cuando esta activo, por medio de una entrada digital del microcontrolador se pueden interpretar estos datos.

##### **Sensor fotovoltaico**

Es una resistencia LDR, el objetivo es capturar las variaciones que esta pueda sufrir con el fin de determinar la cantidad de luz presente en una habitación, la señal que envía esta resistencia es analógica por lo tanto se debe conectarla a la entrada analógica del microcontrolador.

Una salida del microcontrolador conectada a un led de estado con el fin que informe cuando es de día o cuando es de noche dependiendo de la cantidad de luz presente y de lo que se defina en la programación.

#### **4.3.2. MÓDULO XBEE2**

Para el diseño del módulo Xbee2 se usa el PIC 18F452 de gama alta tiene gran capacidad de memoria en relación a los de gama baja.

Un módulo Xbee series 2 conectado a los puertos seriales del microcontrolador RC6 TX y RC7 RX para establecer comunicación con el coordinador.

Dos Leds de estado conectados al módulo Xbee para verificar si esta encendido y escuchando.

## Diagrama de Bloques “XBEE2”

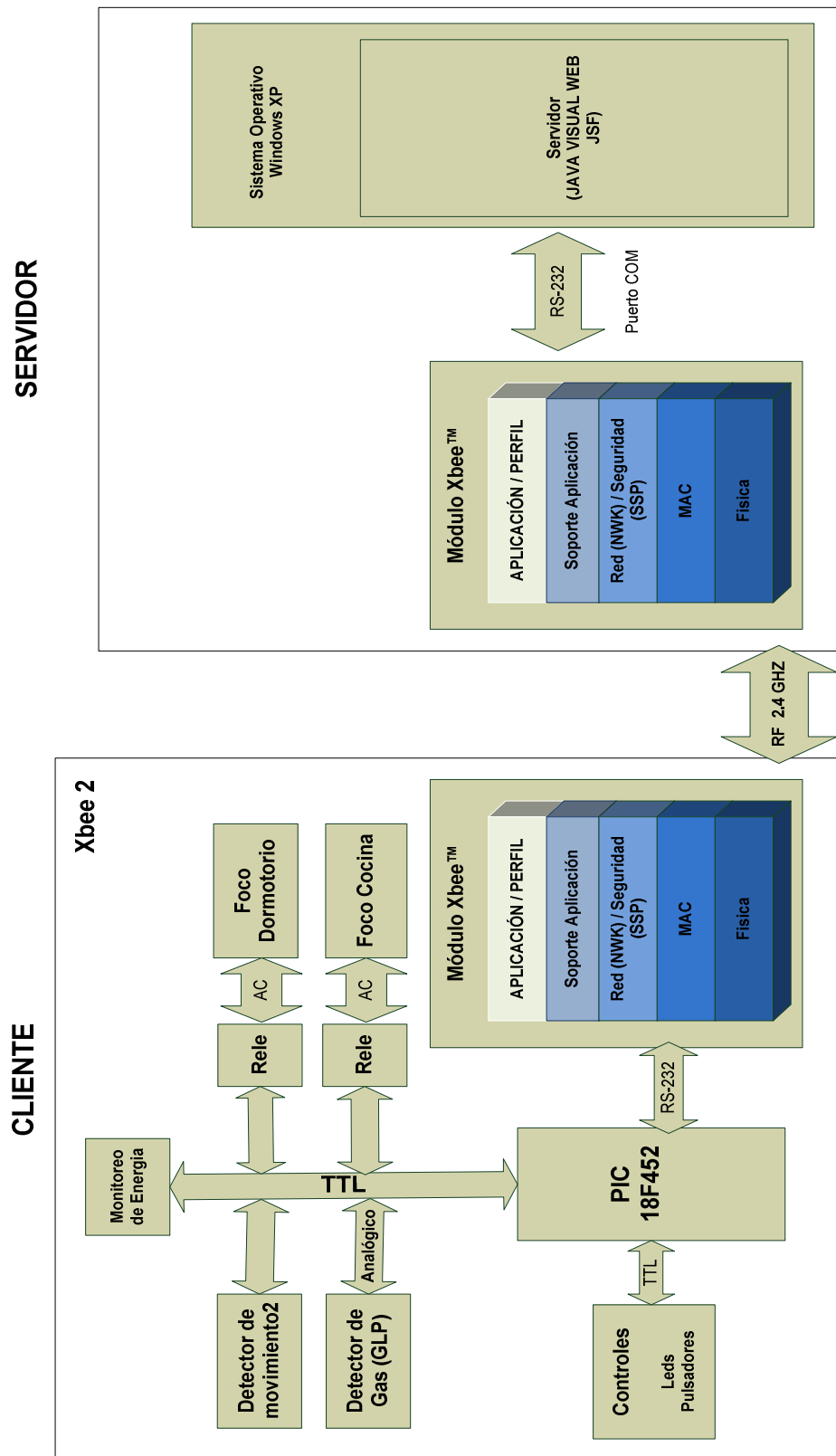


Figura 27: Diagrama de Bloques Módulo Xbee 2

#### **4.3.2.1. Actuadores**

##### **Foco Dormitorio**

Desde una salida del microcontrolador, se conecta un transistor que activa un relé de 12v DC y este a su vez activa el Foco del dormitorio de 110v AC.

Usando otra salida del microcontrolador se activa un led de estado que indica si el foco esta encendido o apagado.

##### **Foco Cocina**

Desde una salida del microcontrolador, se conecta un transistor que activa un relé de 12v DC y este a su vez activa el Foco de la cocina de 110v AC.

Usando otra salida del microcontrolador se activa un led de estado que indica si el foco esta encendido o apagado.

#### **4.3.2.2. Sensores**

##### **Detector de gas GLP**

El detector de gas GLP MQ-6 envía datos analógicos por lo tanto, se debe conectarlo a la entrada analógica del microcontrolador para interpretar estos datos con la programación, y definir bajo que parámetros se activa o desactiva la detección de gas.

Una salida del microcontrolador conectada a un led de estado con el fin que informe cuando se detecte una determinada cantidad de gas GLP en el ambiente.

### **Detector de movimiento**

Envía un pulso bajo cuando esta activado y un pulso alto cuando esta desactivado, está conectado a una entrada digital del microcontrolador.

#### **4.3.3. MÓDULO XBEE3**

Para el diseño del módulo Xbee3 se usa el PIC 18F452 de gama alta tiene gran capacidad de memoria en relación a los de gama baja.

Un módulo Xbee series 2 conectado a los puertos seriales del microcontrolador RC6 TX y RC7 RX para establecer comunicación con el coordinador.

Dos Leds de estado conectados al módulo Xbee para verificar si esta encendido y escuchando.

## Diagrama de Bloques “XBEE3”

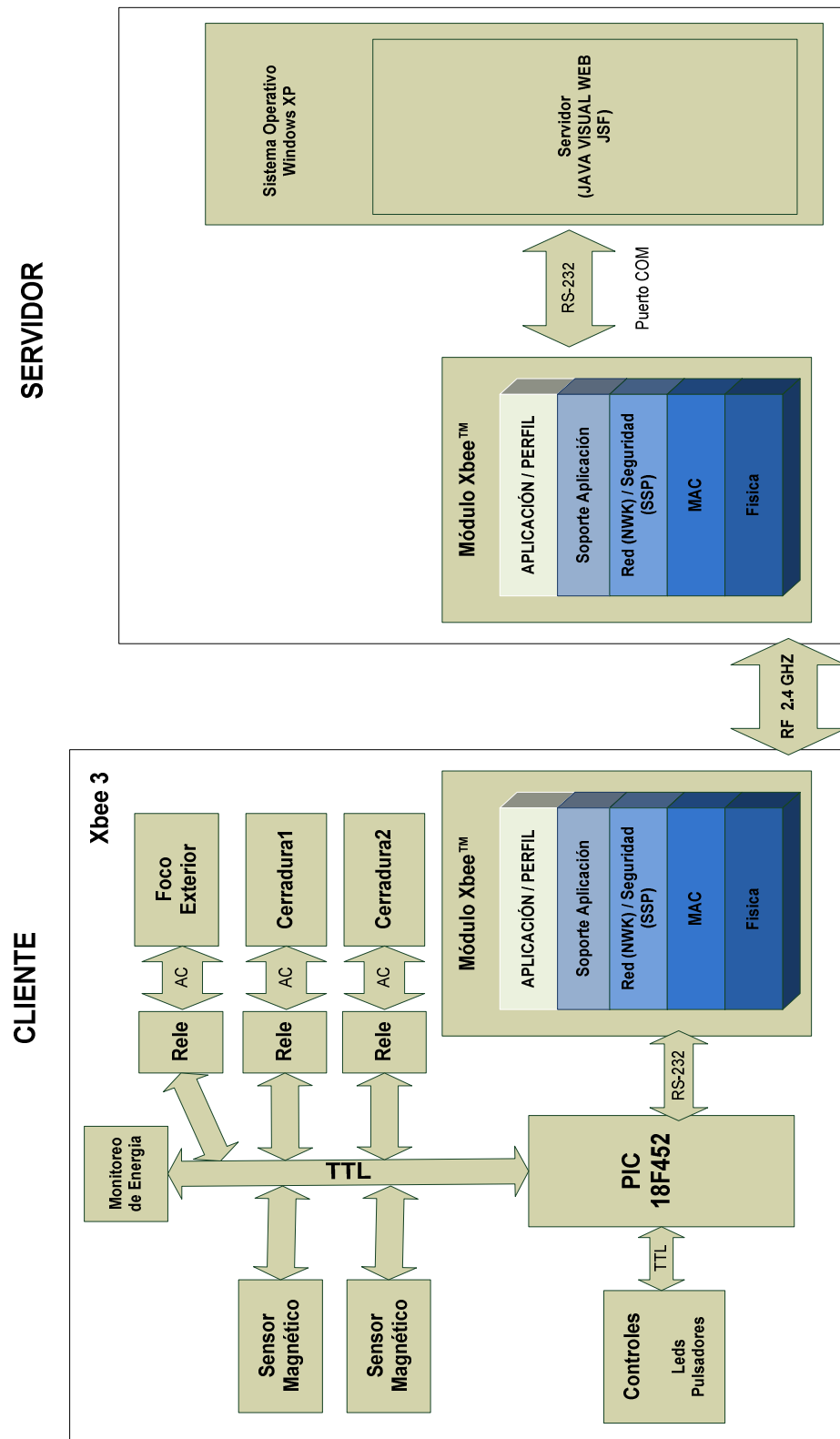


Figura 28: Diagrama de Bloques Módulo Xbee 3

#### **4.3.3.1. Actuadores**

##### **Foco Exterior**

Desde una salida del microcontrolador, se conecta un transistor que activa un relé de 12v DC y este a su vez activa el Foco del exterior de 110v AC.

Usando otra salida del microcontrolador se activa un led de estado que indica si el foco esta encendido o apagado.

##### **Cerradura Eléctrica**

Desde una salida del microcontrolador, se conecta un transistor que activa un relé de 12v DC y este a su vez activa la cerradura de 110v AC.

El pulso que se envía desde el microcontrolador debería ser solo de unos segundos porque las cerraduras eléctricas solo necesitan un corto pulso para activar su bobina y abrir.

Usando otra salida del microcontrolador se activa un led de estado que indica si la cerradura esta activada o desactivada.

#### **4.3.3.2. Sensores**

##### **Sensor Magnético**

Este tipo de sensor da una señal baja si esta desactivado es decir, si no está cerca de su imán, se puede interpretar como puerta abierta, si esta cerca del imán envía una señal alta esto se puede interpretar como puerta cerrada.

Conectado a una entrada digital del microcontrolador para que se interpreten estas señales.

#### **4.3.4. MÓDULO GLCD**

Para el diseño del módulo GLCD se usa el PIC 18F452 de gama mejorada tiene gran capacidad de memoria en relación a los de gama baja, para una aplicación grafica se requiere memoria adicional para almacenar las imágenes que serán presentadas por el GLCD.

El GLCD es de 240 x 128 pixeles con un controlador Toshiba T6963C, permite tener una resolución ideal para la realización de este proyecto, se pueden presentar caracteres y gráficos al usuario, sobrepuesto un panel táctil resistivo permite capturar las coordenadas -X, X, -Y, Y de presión ejercida sobre él, lo que lo convierte en una pantalla táctil.

#### **GLCD + Panel táctil = Pantalla táctil**

El bus de datos del GLCD va conectado al puerto D del microcontrolador D0 – D7.

Los pines de control conectados al puerto B del microcontrolador B1-B3 Y B5.

Los pines del panel táctil son 4, conectados a los pines analógicos del puerto A del microcontrolador en este caso A0 - A3.

El control del back light será manejado por un potenciómetro de 10K.

Leds de estado para verificar si el circuito esta encendido o cargando.

Un módulo Xbee series 2 conectado a los puertos seriales del microcontrolador RC6 TX y RC7 RX para establece comunicación con el coordinador.

Dos Leds de estado conectados al módulo Xbee para verificar si esta encendido y escuchando.

## Diagrama de Bloques “Control remoto inalámbrico local”

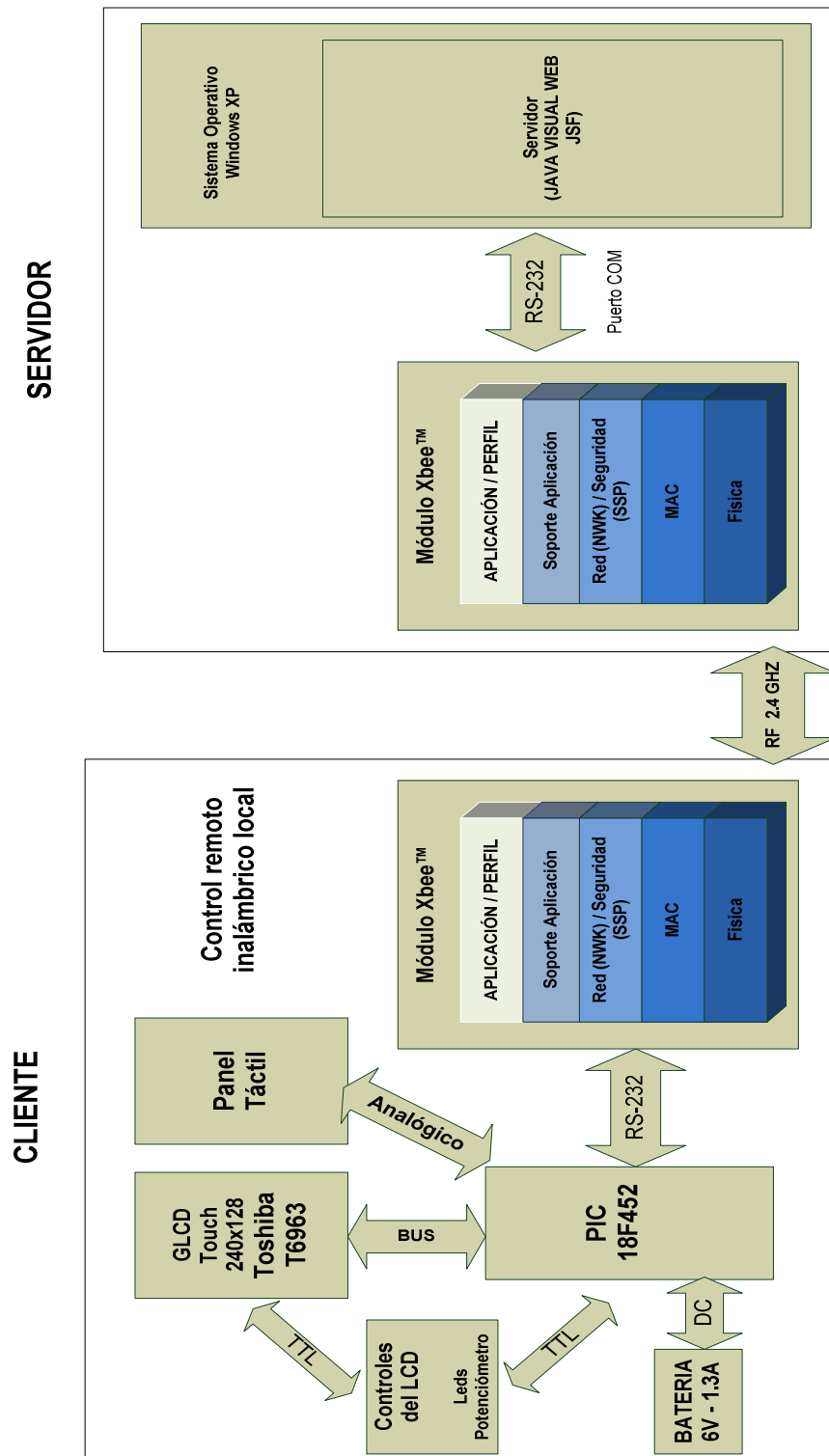


Figura 29: Diagrama de Bloques Módulo GLCD



#### 4.3.5. MÓDULO COORDINADOR

Para conectar un Xbee series 2 al servidor es necesario una interfaz de comunicación en este caso el Xbee Explorer USB.

Se podría construir pero no es recomendable, porque económicamente es más barato comprarlo que diseñar la interfaz USB, además los componentes miniaturizados como el FT232RL dificultan su fabricación.

Se coloca el módulo Xbee series 2 en el explorador USB y esté conectado al computador, se instalan sus respectivos drivers y se maneja de manera serial usando un puerto COM, gestionado por el XCTU o un Hyperterminal.

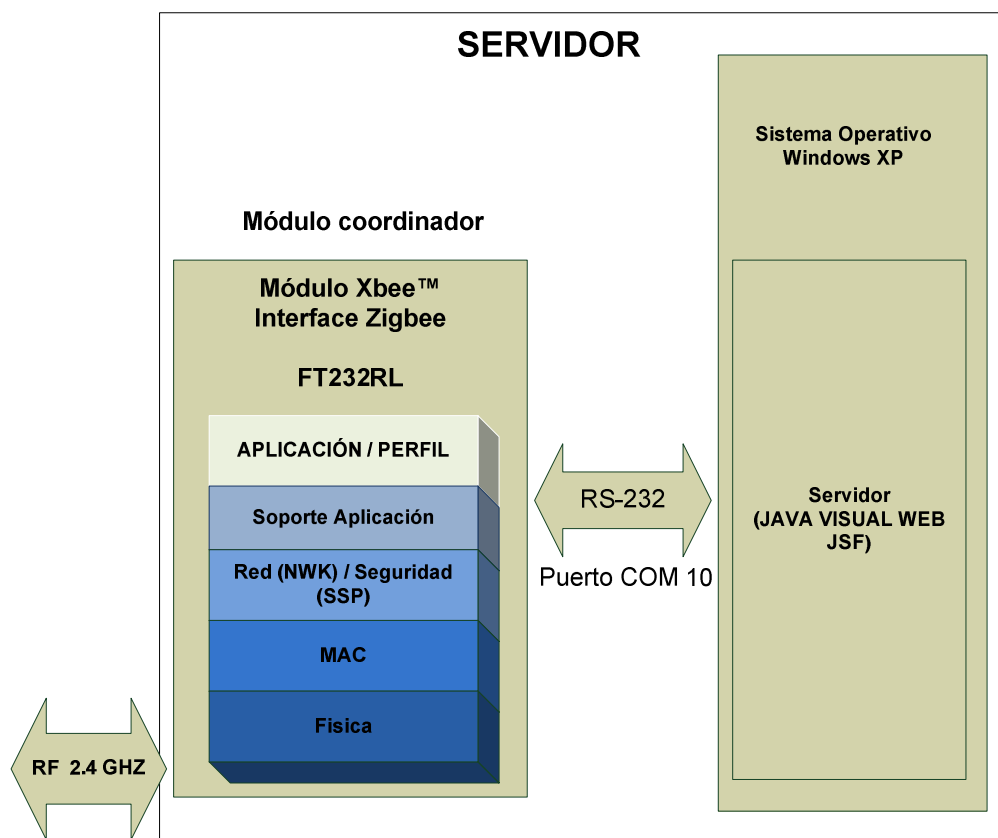


Figura 30: Diagrama de bloques Módulo Coordinador

#### **4.4. DISEÑO DE BASE DE DATOS**

Para el diseño de la base de datos se creó 10 tablas relacionadas entre sí, para que sirvan de medio de almacenamiento de históricos, registro de usuario mediante contraseñas, tipos de usuarios, estados de los módulos Xbee, fechas, horas etc.

También almacena los códigos de comunicación creados para saber exactamente cual módulo Xbee realizó determinada acción, también aquí se tendrá almacenado el número celular de cada usuario, el cual le servirá para interactuar con el sistema, caso contrario al no estar registrado en la base de datos no tendrá acceso.

El mail que se guarde en la base de datos también será al que se envíen las notificaciones resultantes de los eventos del sistema.

Como tablas auxiliares se creó 2 Vistas que almacenarán de forma más ordenada y clara la informaron de los Históricos, tanto del Historial en general como el Historial de los Modos. Con estas vistas se consiguió presentar la información relacionada en el servidor Web de forma más comprensiva y clara.

En los puntos siguientes, se muestran los modelos físico, conceptual de la base de datos ya creada con sus respectivas relaciones.

#### 4.4.1. MODELO FÍSICO DE LA BASE DE DATOS

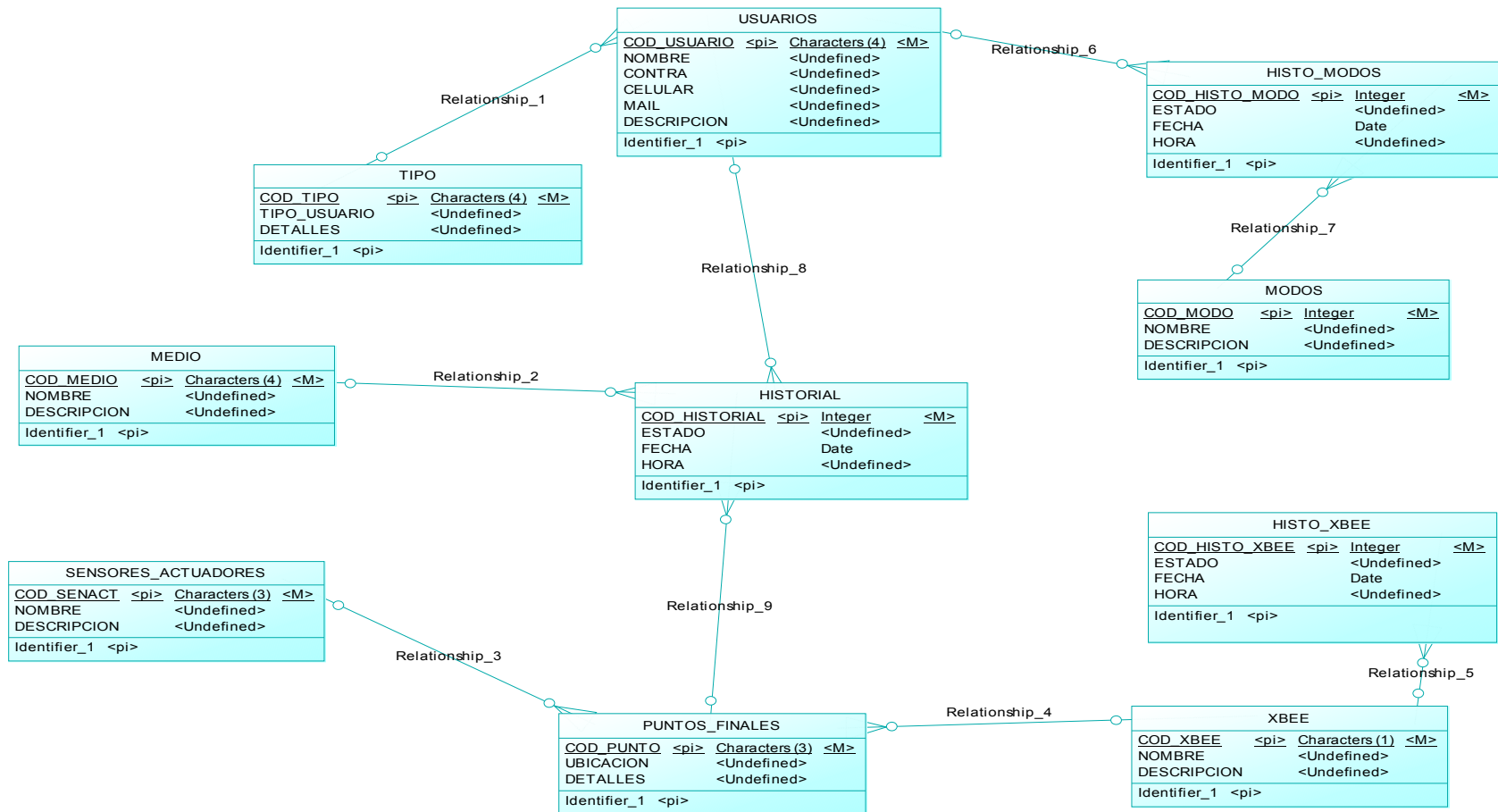


Figura 31: Modelo Físico de la Base de datos

#### 4.4.2. MODELO CONCEPTUAL DE LA BASE DE DATOS

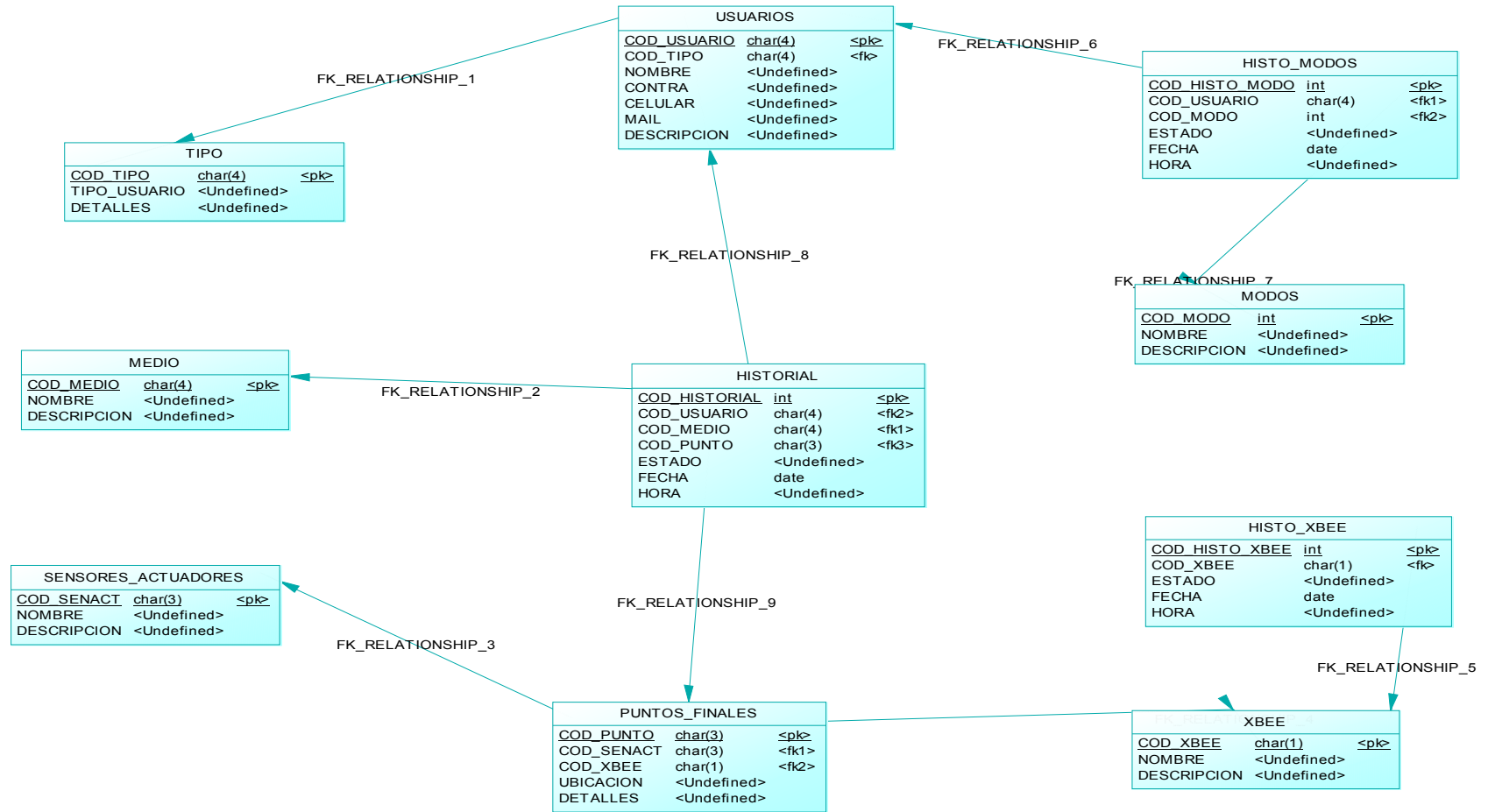
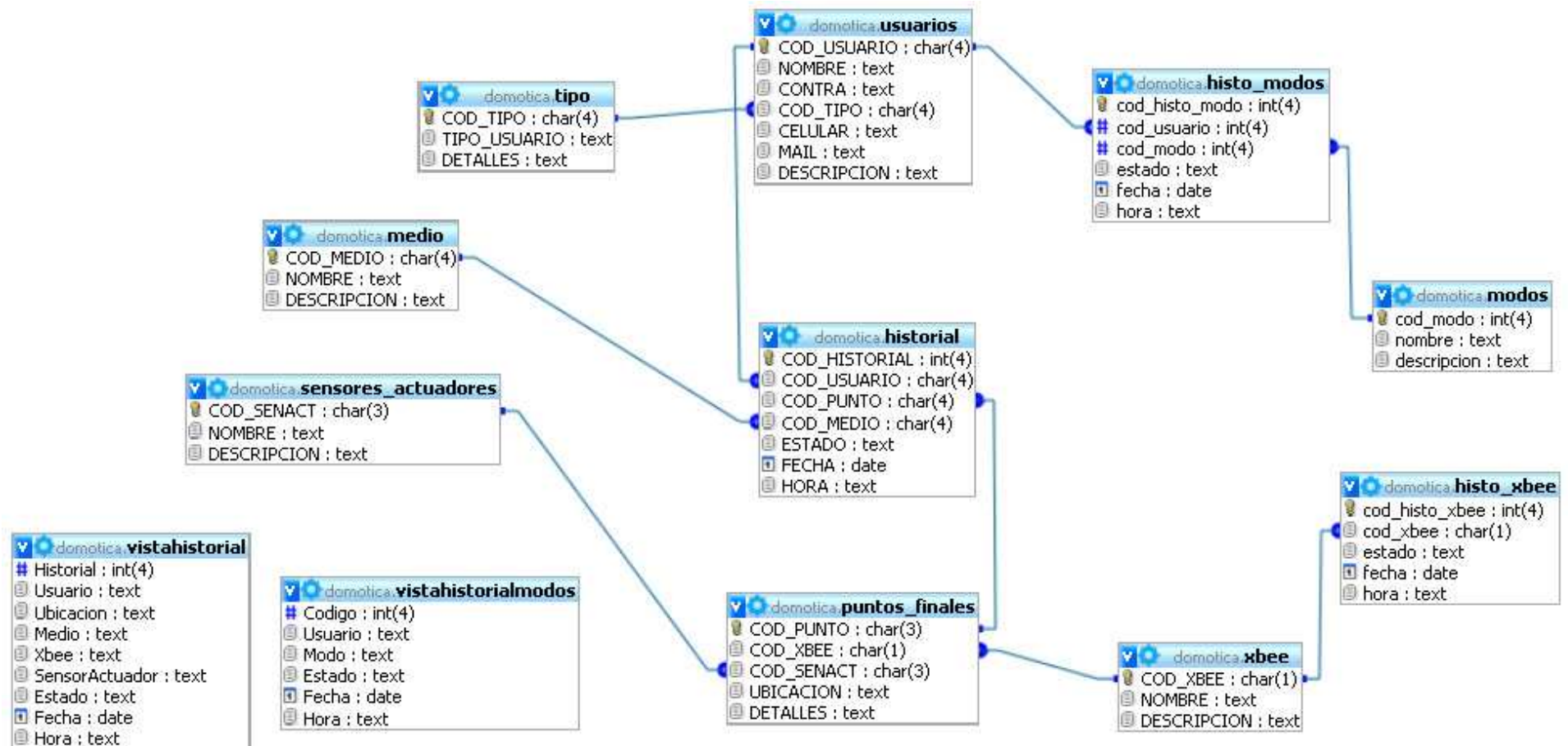


Figura 32: Modelo Conceptual de la Base de datos

#### 4.4.3. DIAGRAMA BASE DE DATOS



**Figura 33: Diagrama de la Base de datos**

#### 4.4.4. DICCIONARIO DE DATOS

Este diccionario de datos mostrará los campos, función, y rol que desempeña cada una de las tablas de la base de datos, para tener una idea general de cómo se realizó su diseño.

**Nombre Tabla:** Usuarios

**Descripción:** Se almacenan los usuarios que tendrán acceso al sistema, su información, y tipo de usuario.

Campo	Tipo	Nulo
COD_USUARIO	char(4)	No
COD_TIPO	char(4)	Si
NOMBRE	char(10)	Si
CONTRA	char(10)	Si
CELULAR	char(10)	Si
MAIL	char(10)	Si
DESCRIPCION	char(10)	Si

**Nombre Tabla:** Tipo

**Descripción:** Se almacena los dos tipos de usuario del sistema, administrador e invitados, éste último tiene limitaciones de acceso.

Campo	Tipo	Nulo
COD_TIPO	char(4)	No
TIPO_USUARIO	char(10)	Si
DETALLES	char(10)	Si

**Nombre Tabla:** Medio

**Descripción:** Aquí se almacena los diferentes medios por los cuales el sistema trabaja: Web, Bluetooth, Control Touch, Físico y SMS. Ayuda a distinguir porque medio se realizó el evento.

Campo	Tipo	Nulo
COD_MEDIO	char(4)	No
NOMBRE	char(10)	Si
DESCRIPCION	char(10)	Si

**Nombre Tabla:** Xbee

**Descripción:** Se almacenan los 4 Xbee que gestionan los módulos mediante ésta tabla se puede conocer que Xbee realizó la acción.

Campo	Tipo	Nulo
COD_XBEE	char(1)	No
NOMBRE	char(10)	Si
DESCRIPCION	char(10)	Si

**Nombre Tabla:** sensores\_actuadores

**Descripción:** Esta tabla contiene todos los sensores y actuadores que el sistema gestiona, con su respectiva descripción.

Campo	Tipo	Nulo
COD_SENACT	char(3)	No
NOMBRE	char(10)	Si
DESCRIPCION	char(10)	Si

**Nombre Tabla:** puntos\_finales

**Descripción:** El campo código de esta tabla es muy importante porque es parte de los códigos de comunicación creados.

Campo	Tipo	Nulo
COD_PUNTO	char(3)	No
COD_SENACT	char(3)	Si
COD_XBEE	char(1)	Si
UBICACION	char(10)	Si
DETALLES	char(10)	Si

**Nombre Tabla:** modos

**Descripción:** Se almacenan los 3 tipos de modos creados: Modo Simulador de Presencia, Modo Ahorro de Energía, Modo Alarma Sonora, para conocer cuál de éstos entro en funcionamiento.

Campo	Tipo	Nulo
COD_MODAL	int	No
NOMBRE	char(10)	Si
DESCRIPCION	char(10)	Si

**Nombre Tabla:** historial

**Descripción:** En esta tabla se almacena toda la información referente a acciones realizadas por el sistema, como es: usuario, punto, medio, fecha, para saber con exactitud quien y en momento realizo determinada acción.

Campo	Tipo	Nulo
COD_HISTORIAL	int	No
COD_USUARIO	char(4)	Si
COD_MEDIO	char(4)	Si
COD_PUNTO	char(3)	Si
ESTADO	char(10)	Si
FECHA	date	Si
HORA	char(10)	Si

**Nombre Tabla:** histo\_xbee

**Descripción:** Aquí se almacena el funcionamiento correcto o no del Xbee para conocer cuando falló la energía en cualquiera de los módulos Xbee

Campo	Tipo	Nulo
COD_HISTO_XBEE	int	No
COD_XBEE	char(1)	Si
ESTADO	char(10)	Si
FECHA	date	Si
HORA	char(10)	Si

**Nombre Tabla:** histo\_modos

**Descripción:** La tabla almacena un historial preciso de quien Activo o Desactivo determinado Modo, se conoce el usuario, la fecha y por su puesto el modo.

Campo	Tipo	Nulo
COD_HISTO_MODO	int	No
COD_USUARIO	char(4)	Si
COD_MODO	int	Si
ESTADO	char(10)	Si
FECHA	date	Si
HORA	char(10)	Si



**Nombre Vista:** vistahistorial

**Descripción:** Esta vista almacena de forma ordenada y clara la información de varias tablas, con esta vista se puede llevar un registro exacto de las acciones realizadas.

Campo	Tipo	Nulo
Historial	int	Si
Usuario	char(4)	Si
Ubicación	char(10)	Si
Medio	char(4)	Si
Xbee	char(4)	Si
SensorActuador	char(10)	Si
Estado	char(10)	Si
Fecha	date	Si
Hora	char(10)	Si

**Nombre Vista:** vistahistorialmodos

**Descripción:** Aquí la vista almacena el historial generado por varias tablas pero hace referencia más a la tabla modos, para mostrar la actividad que se realiza con los diferentes modos.

Campo	Tipo	Nulo
Codigo	int	Si
Usuario	char(4)	Si
Modo	char(10)	Si
Estado	char(10)	Si
Fecha	date	Si
Hora	char(10)	Si

## **4.5. DISEÑO DE SOFTWARE**

### **4.5.1. MICROCONTROLADORES**

#### **4.5.1.1. Módulo Xbee1**

Para el diseño de software del microcontrolador ubicado en el módulo Xbee1 se usa el lenguaje BASIC para su programación, desarrollador MicroCode Studio versión 3.0.0.5 y el compilador PicBasic Pro PBP 2.5 C usando MPASM para compilar los Microcontroladores de gama al 18 F.

MicroCode Studio ofrece gran sencillez y versatilidad al programar, lo que requiere este módulo es controlar las salidas y entradas digitales, los pines de transmisión serial y una entrada analógica.

El microcontrolador inicia escuchando el puerto serial.

Si llega un dato conocido tal como “A”, “H”, “I”, “J” o “x” realiza la función determinada para cada dato reconocido, activar o desactivar una o varias salidas y enviar vía serial el código de estado del punto final afectado.

Dato = “A” Activa/Desactiva puerto D.0 para controlar relé foco sala y envía “1121” o “1120” de manera serial.

Dato = “H” Activa/Desactiva puerto D.1 para controlar relé bocina y envía “1811” o “1810” de manera serial.

Dato = “I” Activa/Desactiva puerto D.2 para controlar relé abre cortina y envía “1911” de manera serial.

Dato = “J” Activa/Desactiva puerto D.2 para controlar relé cierra cortina y envía “1910” de manera serial.

Dato = “x” Envía el estado de los dispositivos finales.

Si no llega un dato conocido o no llega ningún dato al pasar un determinado tiempo, comienza a escuchar sus entradas digitales y analógicas en busca de cambios si sucede alguno realiza la función correspondiente.

#### **4.5.1.1.1. Diagrama de Flujo**

## Diagrama de flujo "Xbee 1"

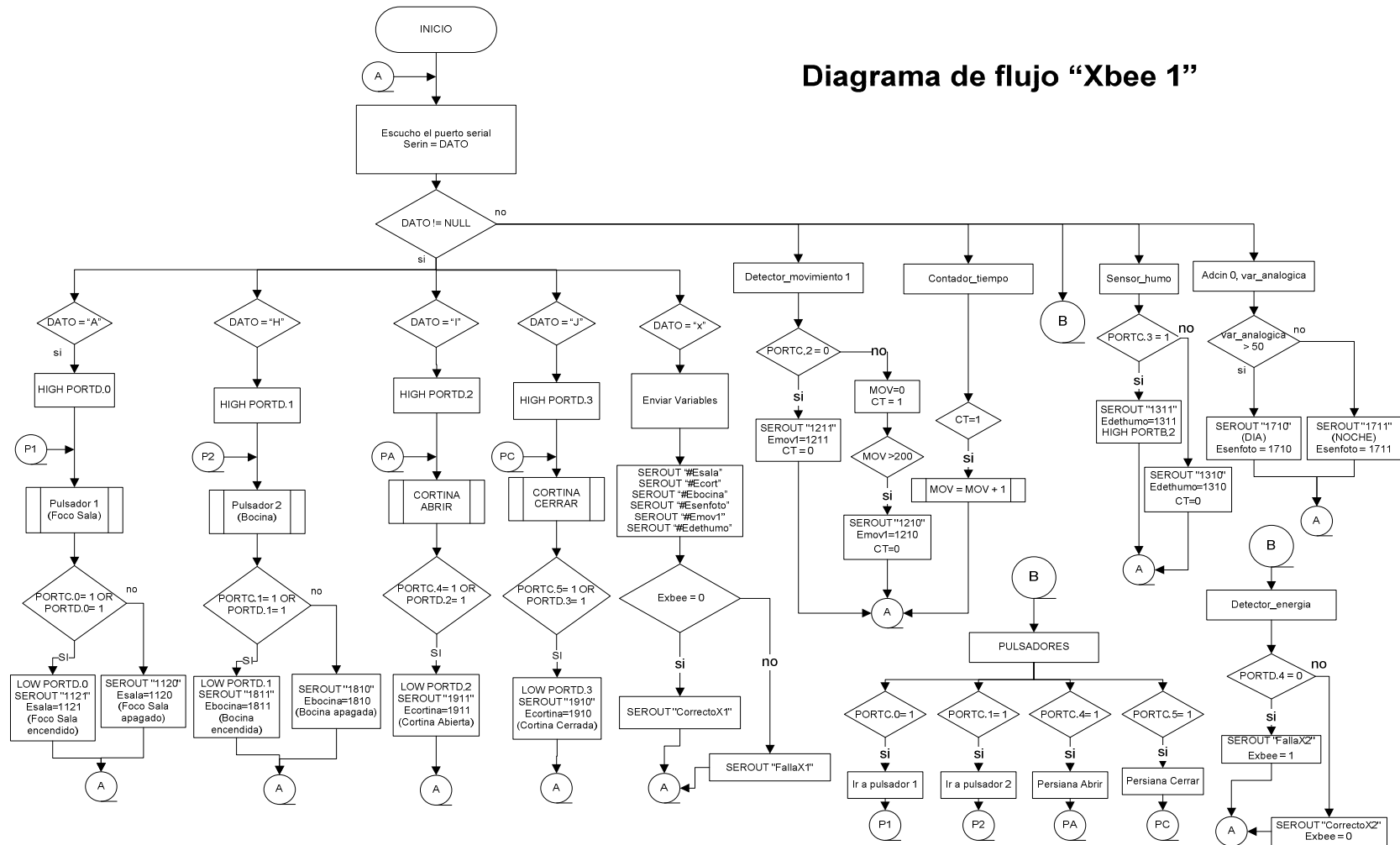


Figura 34: Diagrama de Flujo Módulo Xbee 1

#### **4.5.1.2. Módulo Xbee2**

Para el diseño de software del microcontrolador ubicado en el módulo Xbee2 se usa el lenguaje BASIC para su programación, desarrollador MicroCode Studio versión 3.0.0.5 y el compilador PicBasic Pro PBP 2.5 C usando MPASM para compilar los Microcontroladores de gama al 18 F.

MicroCode Studio ofrece gran sencillez y versatilidad al programar, lo que requiere este módulo es controlar las salidas y entradas digitales, los pines de transmisión serial y una entrada analógica.

El microcontrolador inicia escuchando el puerto serial.

Si llega un dato conocido tal como “C”, “D” o “y” realiza la función determinada para cada dato reconocido , activar o desactivar una o varias salidas y enviar vía serial el código de estado del punto final afectado.

Dato = “C” Activa/Desactiva puerto D.O para controlar relé foco dormitorio y envía “2111” o “2110” de manera serial.

Dato = “D” Activa/Desactiva puerto D.1 para controlar relé foco cocina y envía “2131” o “2130” de manera serial.

Dato = “y” Envía el estado de los dispositivos finales.

Si no llega un dato conocido o no llega ningún dato al pasar un determinado tiempo, comienza a escuchar sus entradas digitales y analógicas en busca de cambios si sucede alguno realiza la función correspondiente.

#### 4.5.1.2.1. Diagrama de Flujo

### Diagrama de flujo "Xbee 2"

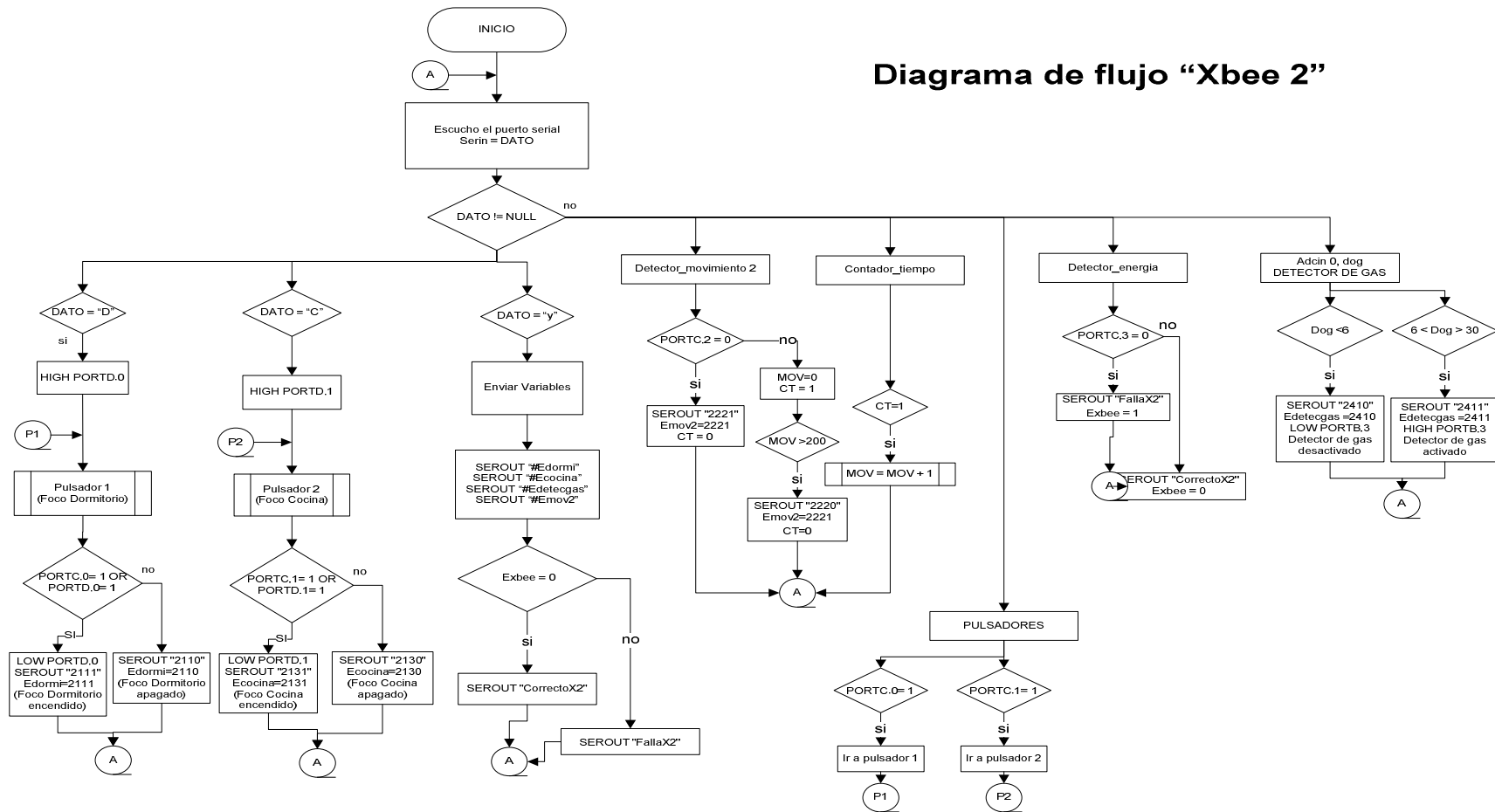


Figura 35: Diagrama de Flujo Módulo Xbee 2

#### **4.5.1.3. Módulo Xbee3**

Para el diseño de software del microcontrolador ubicado en el módulo Xbee3 se usa el lenguaje BASIC para su programación, desarrollador MicroCode Studio versión 3.0.0.5 y el compilador PicBasic Pro PBP 2.5 C usando MPASM para compilar los Microcontroladores de gama al 18 F.

MicroCode Studio ofrece gran sencillez y versatilidad al programar y lo que requiere este módulo es controlar las salidas y entradas digitales Y los pines de transmisión serial.

El microcontrolador inicia escuchando el puerto serial.

Si llega un dato conocido tal como “B”, “F”, “G” o “z” realiza la función determinada para cada dato reconocido, activar o desactivar una o varias salidas.

Dato = “B” Activa/Desactiva puerto D.0 para controlar relé foco exterior y envía “3141” o “3140” de manera serial.

Dato = “F” Activa/Desactiva puerto D.1 para controlar relé de activación de la cerradura eléctrica y envía “3611” de manera serial.

Dato = “G” Activa/Desactiva puerto D.2 para controlar relé de activación de la cerradura eléctrica y envía “3621” de manera serial.

Dato = “y” Envía el estado de los dispositivos finales.

Si no llega un dato conocido o no llega ningún dato al pasar un determinado tiempo, comienza a escuchar sus entradas digitales en busca de cambios si sucede alguno realiza la función correspondiente.

#### 4.5.1.3.1. Diagrama de Flujo

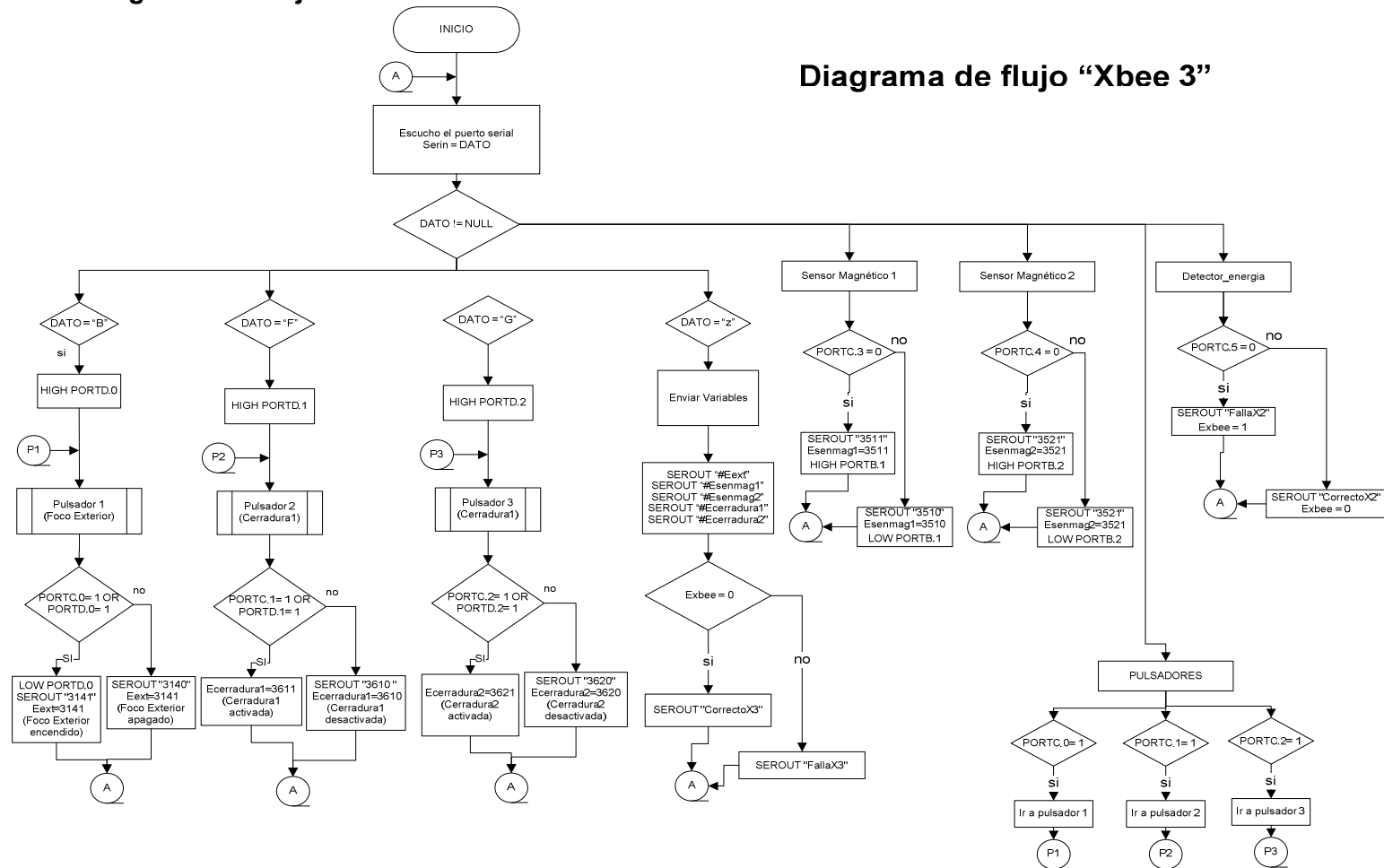


Figura 36: Diagrama de Flujo Módulo Xbee 3

#### 4.5.1.4. Módulo GLCD Touch

Para el diseño de software del microcontrolador ubicado en el módulo GLCD Touch se usa el lenguaje BASIC para su programación, desarrollador MikroBasic versión 7.0.0.2 y el compilador MikroLCD Debugger.

Se manejó dos tipos de usuarios el administrador tiene acceso total y el invitado tiene acceso solo a control de iluminación y monitoreo.



Figura 37: Gráfico de navegación administrador



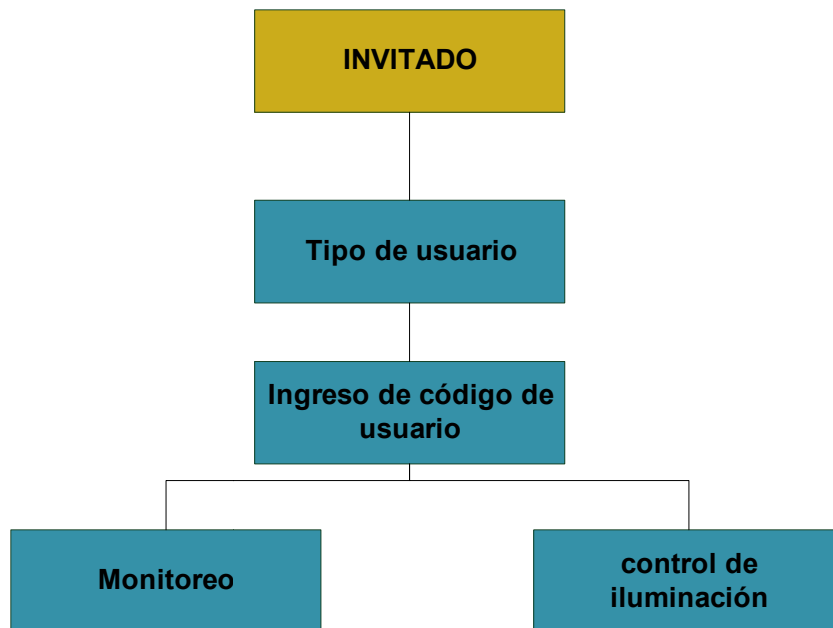


Figura 38: Gráfico de navegación invitado

Está compuesto por 5 pantallas bmp:



Figura 39: Pantalla tipo de usuario

En esta pantalla se puede seleccionar el tipo de usuario que va a ingresar sea Administrador o Invitado, ofrece una función adicional “SOS”, este es un botón de pánico que el usuario podrá activar sin necesidad de iniciar su sesión.

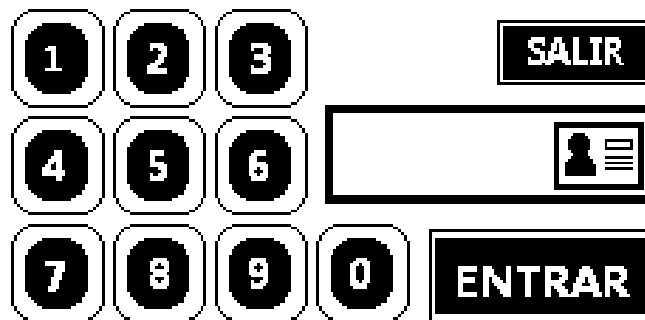


Figura 40: Pantalla ingreso código de usuario

En esta pantalla se digita el código de usuario y a continuación, clic en entrar, si esta correcto ingresa, si no vuelve a pedir el código una vez más, tiene 3 posibilidades si las excede el sistema se bloquea por un tiempo determinado.

Pestañas para gestionar la iluminación, monitoreo de sensores y cerraduras.

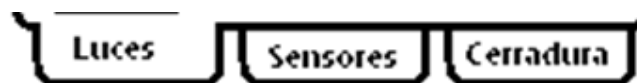


Figura 41: Pestañas para navegación

Para navegar entre las diferentes pestañas el usuario debe hacer clic sobre la que necesite ver.



Figura 42: Pestaña control de iluminación

En esta pestaña aparece el estado inicial de cada foco y de los modos ahorro de energía y simulador de presencia, el estado viene dado por ON / OFF al lado derecho de cada botón, para gestionar un foco o un modo basta con presionar el botón, envía una señal al coordinador y este al respectivo módulo Xbee en donde se encuentre el dispositivo final a ser gestionado.

El botón Salir cierra la sesión y reinicia el software del control remoto inalámbrico.

Esta pestaña está disponible para todos los tipos de usuarios.

MONITOREO DE SENSORES	
Puerta 1	<input type="text"/>
Puerta 2	<input type="text"/>
Movim 1	<input type="text"/>
Movim 2	<input type="text"/>
Detc. Humo	<input type="text"/>
Detc. Gas	<input type="text"/>
Detc. Luz	<input type="text"/>

Luces    Sensores    Cerradura

Figura 43: Pestaña Monitoreo de sensores

En esta pestaña se monitorea los sensores de nuestro sistema domótico se actualiza cada vez que se entra a ella.

Se monitorean los sensores magnéticos ubicados en las puertas, detectores de movimiento, gas, humo y el detector de luz que informa si es día o noche.

El botón Salir cierra la sesión y reinicia el software del control remoto inalámbrico.

Esta pestaña está disponible para todos los tipos de usuarios.

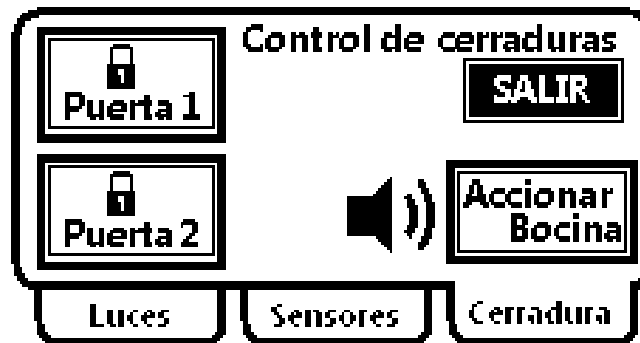


Figura 44: Pestaña control de cerraduras

En esta pestaña se controla la parte sensible de la casa como son las cerraduras, son pulsos que se envían a cerraduras eléctricas ubicadas en las puertas para abrirlas. Se debe presionar sobre el botón puerta 1 o puerta 2.

Una aplicación adicional es el botón accionar bocina, también es posible mandar un pulso para activar la bocina con el fin de probarla o enviar una alerta sonora ante cualquier eventualidad.

El botón Salir cierra la sesión y reinicia el software del control remoto inalámbrico.

Esta pestaña no está disponible para todos los tipos de usuarios solo para el tipo Administrador.

#### 4.5.1.4.1. Diagrama de Flujo

##### Diagrama de flujo “ Control remoto inalámbrico local “

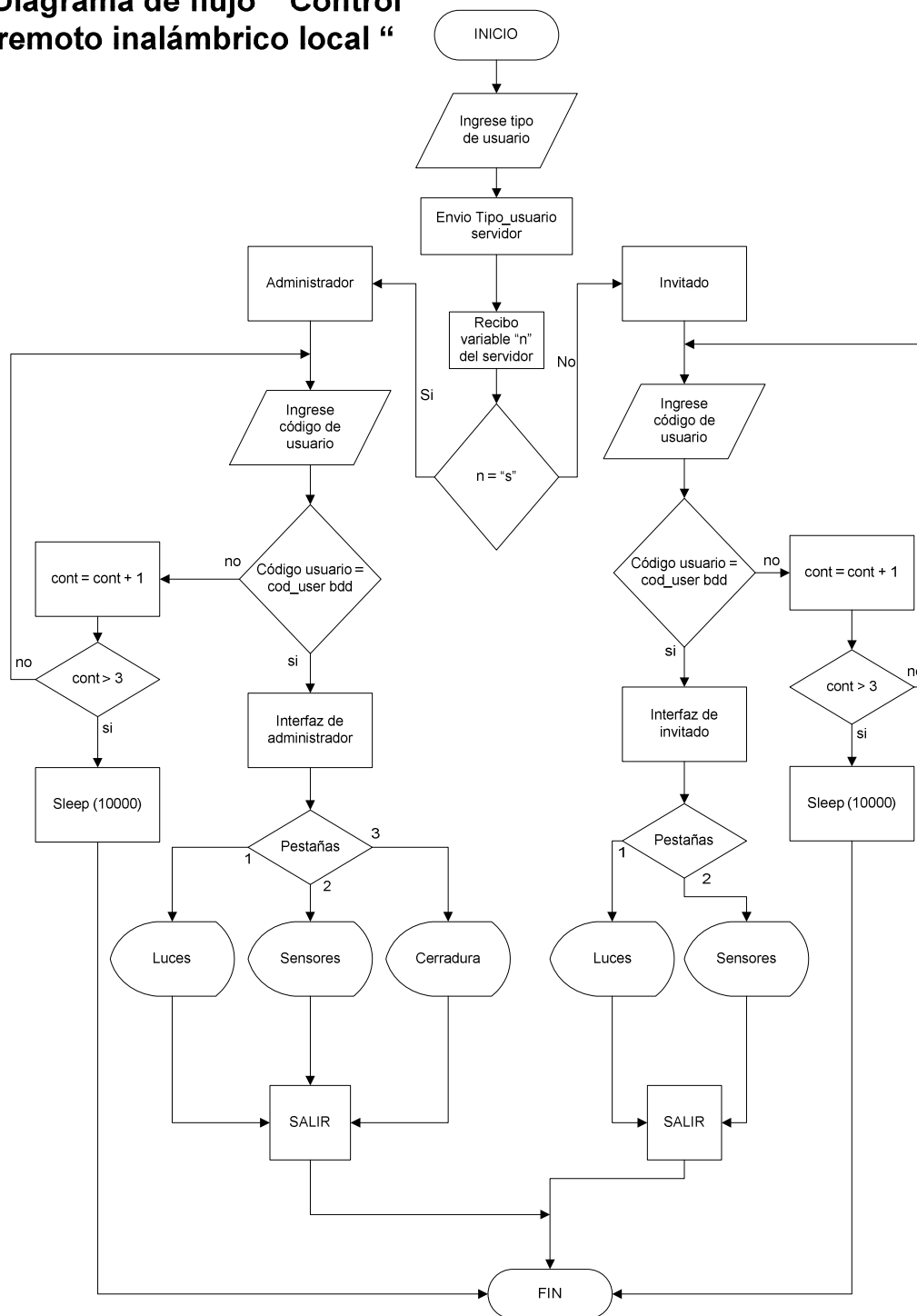


Figura 45: Diagrama de Flujo GLCD

#### 4.5.1.4.2. Casos de uso

##### CASOS DE USO “ Control remoto inalámbrico local ”



Figura 46: Casos de uso GLCD

Caso de uso 01	
<b>Nombre:</b>	Envío Emergencia
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Enviar vía serial un llamado de Emergencia
<b>Descripción:</b>	Se envía una alerta de Emergencia al servidor con el fin que se informe al administrador.
<b>Objetivo</b>	Simular un botón de pánico, notificando de este evento al administrador.

<b>Caso de uso 02</b>	
<b>Nombre:</b>	Comunicación Serial con el servidor principal
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Envía o recibe el dato solicitado vía serial hacia y desde el servidor
<b>Descripción:</b>	Por medio del puerto COM del Xbee GLCD se envía el dato al puerto COM del servidor y viceversa
<b>Objetivo</b>	Establecer un canal de comunicación serial entre el Xbee GLCD y el Xbee Coordinador.

<b>Caso de uso 03</b>	
<b>Nombre:</b>	Módulo Xbee
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe dato, lo interpreta y responde
<b>Descripción:</b>	Interpreta los datos recibidos por el cliente GLCD y responde a las peticiones luego de preguntar a los módulos lo que necesite.
<b>Objetivo</b>	Recibir y enviar datos.

<b>Caso de uso 04</b>	
<b>Nombre:</b>	Consultar BDD
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Realizar consultas a la base de datos
<b>Descripción:</b>	Realiza consultas requeridas por el usuario a la base de datos
<b>Objetivo</b>	Consultar base de datos

<b>Caso de uso 05</b>	
<b>Nombre:</b>	Seleccionar el tipo de usuario e ingresar el código
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe tipo de usuario y código

<b>Descripción:</b>	Almacena los datos recibidos en variables para ser enviados a comprobación.
<b>Objetivo</b>	Almacenar tipo de usuario y código

<b>Caso de uso 06</b>	
<b>Nombre:</b>	Comprobar identidad del usuario
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Verifica los datos recibidos
<b>Descripción:</b>	Compara los datos recibidos por ingreso, con los que están almacenados en la base de datos
<b>Objetivo</b>	Comprobar identidad de usuario

<b>Caso de uso 07</b>	
<b>Nombre:</b>	Control Iluminación
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Controla la iluminación
<b>Descripción:</b>	Envía los códigos de comunicación que controlan la iluminación.
<b>Objetivo</b>	Controlar la iluminación vía GLCD

<b>Caso de uso 08</b>	
<b>Nombre:</b>	Módulo SMS
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe dato, lo interpreta y realiza una acción
<b>Descripción:</b>	Interpreta los datos recibidos por el cliente GLCD y luego realiza acciones específicas como enviar un SMS
<b>Objetivo</b>	Recibir interpretarlos y realizar una acción específica



<b>Caso de uso 09</b>	
<b>Nombre:</b>	Módulo Correo
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Interacción con la cuenta de correo
<b>Descripción:</b>	Envía correos electrónicos a direcciones especificadas
<b>Objetivo</b>	Enviar correo electrónico.

<b>Caso de uso 10</b>	
<b>Nombre:</b>	Monitoreo de sensores
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Pedir estado de sensores
<b>Descripción:</b>	Realiza peticiones del estado de los sensores que requiera.
<b>Objetivo</b>	Recibir el estado de los sensores.

<b>Caso de uso 11</b>	
<b>Nombre:</b>	Control de Cerraduras
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar las cerraduras eléctricas
<b>Descripción:</b>	Realiza peticiones para activar las cerraduras que se necesite.
<b>Objetivo</b>	Activar cerraduras

#### 4.5.1.4.3. Diagrama de clases (módulos)

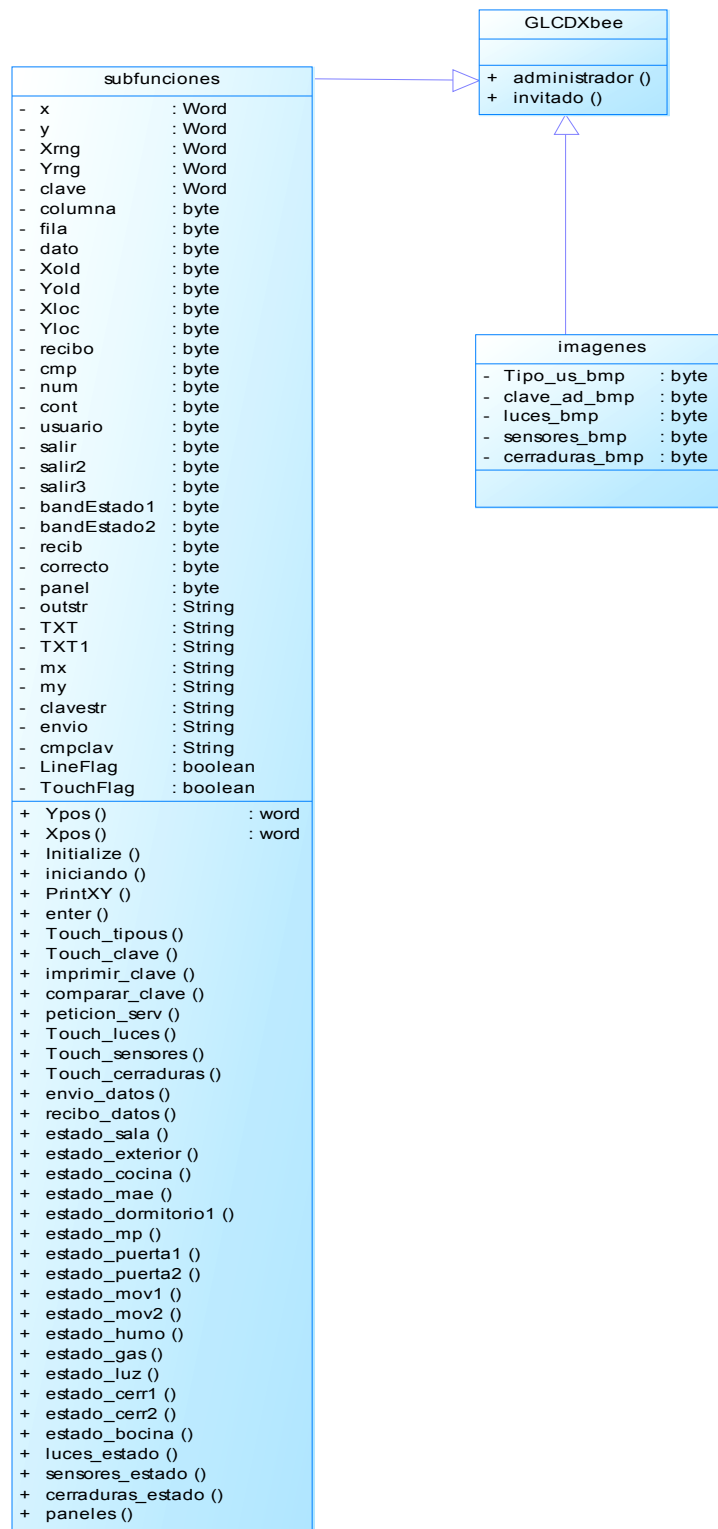


Figura 47: Diagrama de Clases GLCD

#### **4.5.2. APLICACIÓN MÓVIL BLUETOOTH J2ME**

Para el desarrollo de la aplicación Bluetooth se necesita escribir un programa en J2ME, que se instale como un JAR en el teléfono celular y pueda ejecutarse, el objetivo es generar un puerto COM con el cual se pueda establecer comunicación con otro puerto COM virtual, generado en el computador a partir de un dispositivo Bluetooth USB.

Para escribir programas en Java 2 Micro Edition J2ME se usa el Wireless Toolkit de Sun WTK, pero existe un desarrollador que facilita aún mucho más la compilación de estos programas es el Mobile Processing 007.

Por medio de la librería `processing.bluetooth` se descubren dispositivos, se selecciona uno de ellos y si acepta la petición se establece una comunicación entre sus puertos seriales, COM del dispositivo bluetooth móvil y COM del dispositivo Bluetooth USB.

Y se hace uso de la capa RFCOMM de Bluetooth estableciendo una comunicación directa vía serial desde el Hyperterminal del computador hacia el dispositivo móvil.

La aplicación está diseñada para un dispositivo móvil con pantalla de 176x220 pixeles (Sony Ericsson K550i).

La aplicación está compuesta por 5 pantallas.

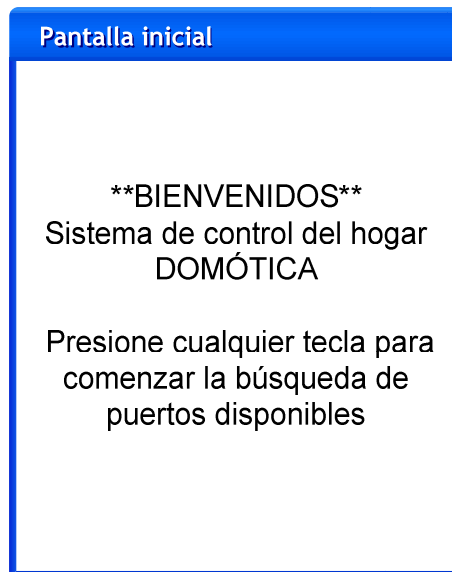


Figura 48: Pantalla inicial Aplicación Bluetooth

Da la bienvenida a la aplicación bluetooth al presionar cualquier tecla comienza la búsqueda de puertos disponibles entre los dispositivos bluetooth cercanos.

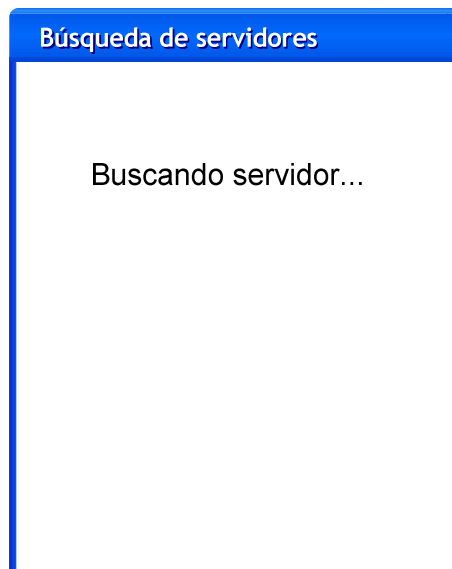
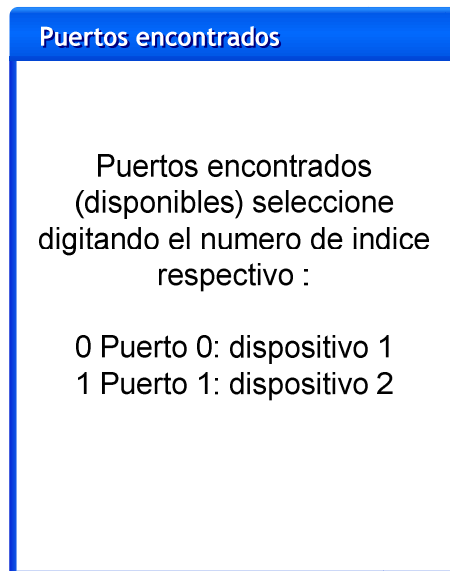


Figura 49: Pantalla Búsqueda de servidores Puertos disponibles

Realiza la búsqueda de puertos seriales disponibles, activa vibración al buscar.



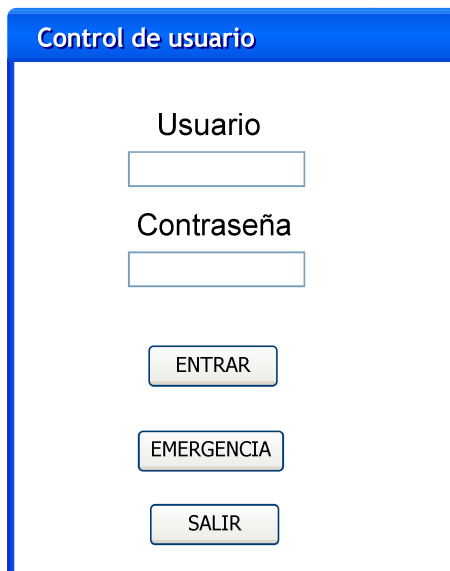
Puertos encontrados

Puertos encontrados  
(disponibles) seleccione  
digitando el numero de indice  
respectivo :

0 Puerto 0: dispositivo 1  
1 Puerto 1: dispositivo 2

Figura 50: Pantalla Puertos encontrados

Lista los puertos encontrados, para seleccionar el puerto correspondiente al servidor digitar el índice respectivo.



Control de usuario

Usuario

Contraseña

ENTRAR

EMERGENCIA

SALIR

Figura 51: Pantalla inicial Control de usuario

El botón “entrar” realiza el reconocimiento del usuario comunicándose con el servidor, si esta correcto avanza a la siguiente pantalla, el botón emergencia brinda el servicio de botón de pánico, informa al servidor que algo que está sucediendo para que realice tareas de notificaciones. El botón salir cierra la aplicación.



Figura 52: Pantalla Control de iluminación y monitoreo Aplicación Bluetooth

Esta pantalla se presenta una vez que el usuario ha sido reconocido por el servidor.

Control de iluminación por medio de los botones Sala, Exterior, Cocina y Dormitorio, al presionar uno de ellos se envía la petición al servidor y este realiza la tarea determinada.

Monitoreo por medio del botón Act. Sensores, muestra el estado de los sensores de movimiento o magnéticos cada vez que se presiona, envía la petición al servidor y este le devuelve el estado de los diferentes sensores.

#### 4.5.2.1. Diagrama de bloques

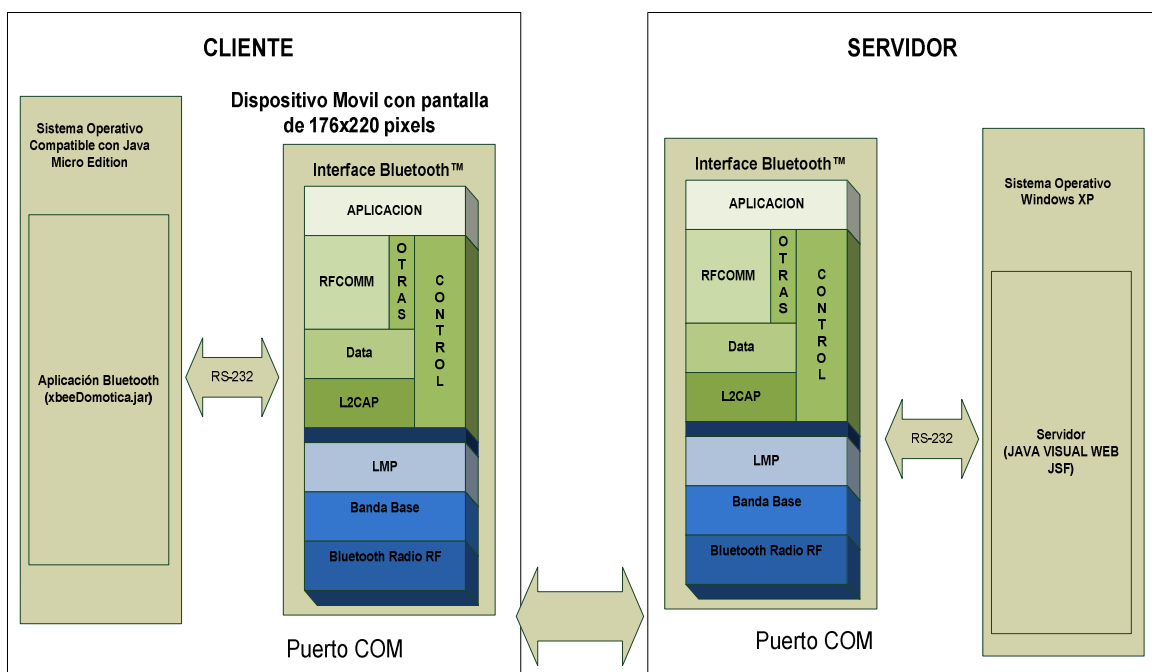


Figura 53: Diagrama de Bloques Aplicación Bluetooth

#### 4.5.2.2. Diagrama de Flujo

### Diagrama de flujo “Aplicación Bluetooth”

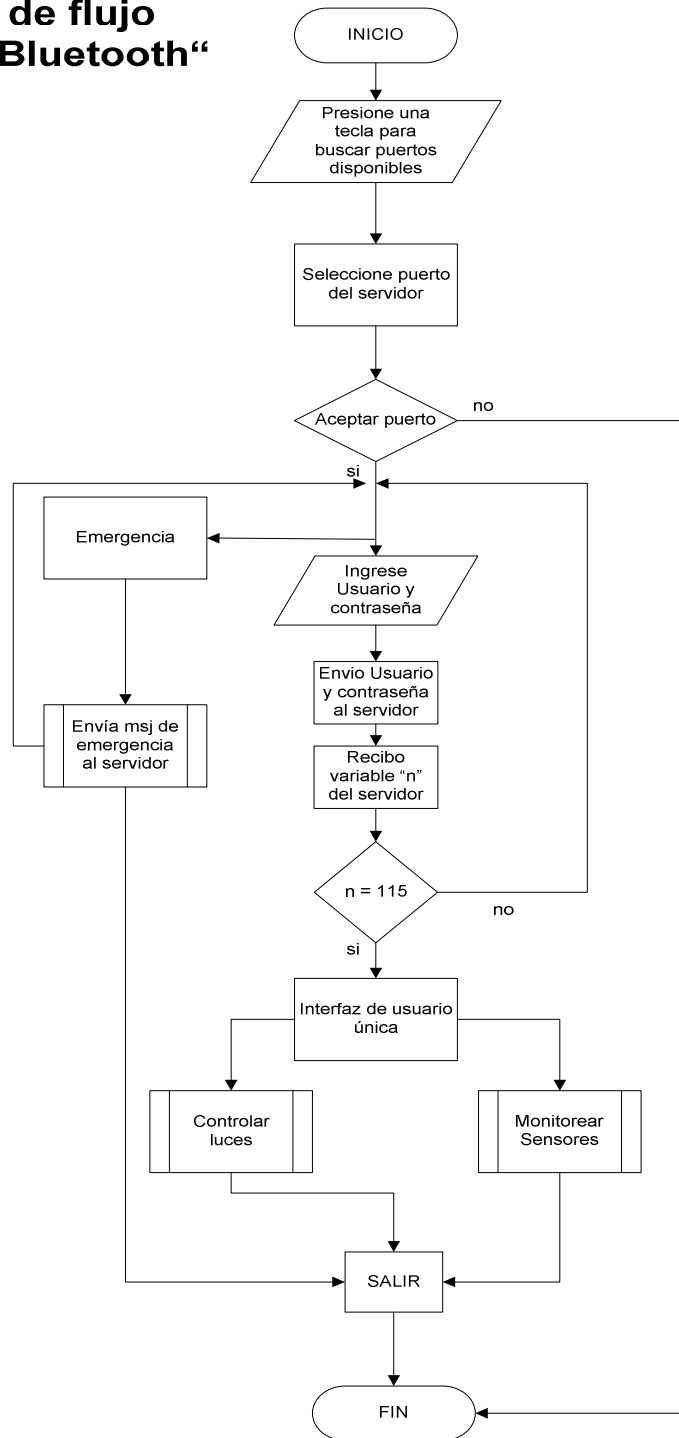


Figura 54: Diagrama de Flujo Aplicación Bluetooth



### 4.5.2.3. Casos de uso

#### CASOS DE USO “ Aplicación Bluetooth “

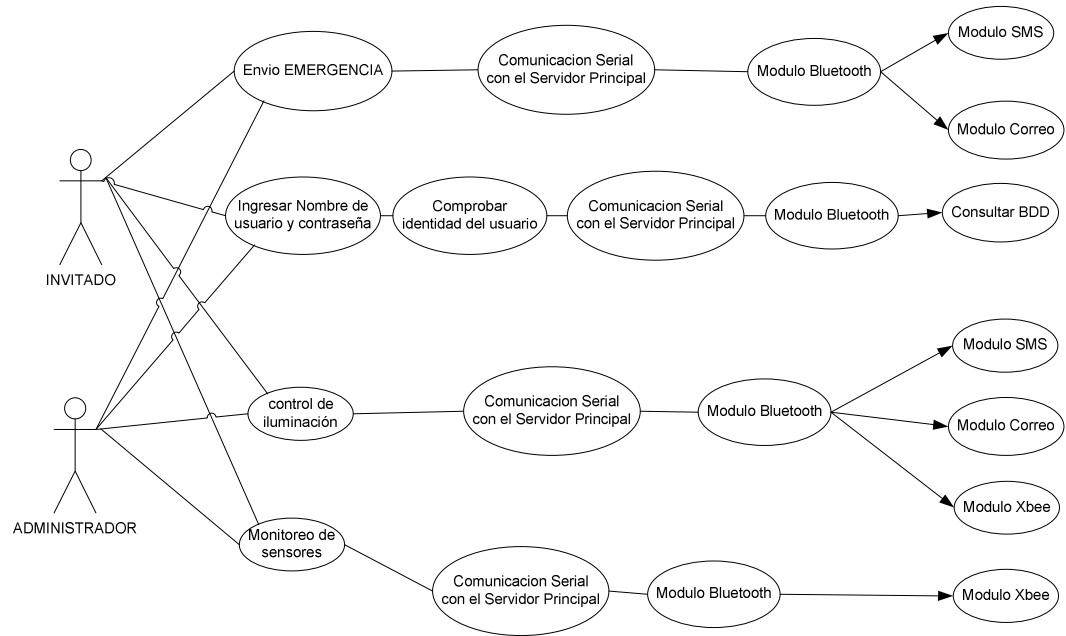


Figura 55: Casos de uso Aplicación Bluetooth

Caso de uso 01	
<b>Nombre:</b>	Envío Emergencia
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Enviar vía serial un llamado de Emergencia
<b>Descripción:</b>	Se envía una alerta de Emergencia al servidor con el fin que se informe al administrador vía SMS o mail que algo está sucediendo en el hogar
<b>Objetivo</b>	Simular un botón de pánico, notificando de este evento al administrador.

Caso de uso 02	
<b>Nombre:</b>	Comunicación Serial con el servidor principal
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Envía o recibe el dato solicitado vía serial hacia y

	desde el servidor
<b>Descripción:</b>	Por medio del puerto COM del dispositivo Bluetooth móvil se envía el dato al puerto COM virtual del servidor y viceversa
<b>Objetivo</b>	Establecer un canal de comunicación serial entre los dos dispositivos bluetooth

#### **Caso de uso 03**

<b>Nombre:</b>	Módulo Bluetooth
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe dato, lo interpreta y responde
<b>Descripción:</b>	Interpreta los datos recibidos por el cliente Bluetooth y responde a las peticiones luego de preguntar a los módulos lo que necesite.
<b>Objetivo</b>	Recibir y enviar datos.

#### **Caso de uso 04**

<b>Nombre:</b>	Módulo SMS
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe dato, lo interpreta y realiza una acción
<b>Descripción:</b>	Interpreta los datos recibidos por el cliente Bluetooth y luego realiza acciones específicas como enviar un SMS
<b>Objetivo</b>	Recibir interpretarlos y realizar una acción específica

#### **Caso de uso 05**

<b>Nombre:</b>	Módulo Correo
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Interacción con la cuenta de correo
<b>Descripción:</b>	Envía correos electrónicos a direcciones especificadas
<b>Objetivo</b>	Enviar correo electrónico.

<b>Caso de uso 06</b>	
<b>Nombre:</b>	Ingresar Nombre de Usuario y contraseña
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe nombre de usuario y contraseña
<b>Descripción:</b>	Almacena los datos recibidos en variables para ser enviados a comprobación.
<b>Objetivo</b>	Almacenar usuario y contraseña

<b>Caso de uso 07</b>	
<b>Nombre:</b>	Comprobar identidad del usuario
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Verifica los datos recibidos
<b>Descripción:</b>	Compara los datos recibidos por ingreso, con los que están almacenados en la base de datos
<b>Objetivo</b>	Comprobar identidad de usuario

<b>Caso de uso 08</b>	
<b>Nombre:</b>	Consultar BDD
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Realizar consultas a la base de datos
<b>Descripción:</b>	Realiza consultas requeridas por el usuario a la base de datos
<b>Objetivo</b>	Consultar base de datos

<b>Caso de uso 09</b>	
<b>Nombre:</b>	Control Iluminación
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Controla la iluminación
<b>Descripción:</b>	Envía los códigos de comunicación que controlan la iluminación.
<b>Objetivo</b>	Controlar la iluminación vía bluetooth

<b>Caso de uso 10</b>	
<b>Nombre:</b>	Módulo Xbee
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Envía dato, Recibe dato, lo interpreta y realiza una acción
<b>Descripción:</b>	Interpreta los datos recibidos por el cliente Bluetooth y luego realiza acciones específicas como enviar comandos al coordinador Zigbee y recibir códigos del coordinador Zigbee para enviarlos al cliente Bluetooth
<b>Objetivo</b>	Enviar , Recibir datos e interpretarlos para realizar una acción

<b>Caso de uso 11</b>	
<b>Nombre:</b>	Monitoreo de sensores
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Pedir estado de sensores
<b>Descripción:</b>	Realiza peticiones del estado de los sensores que requiera.
<b>Objetivo</b>	Recibir el estado de los sensores.

#### 4.5.2.4. Diagrama de clases

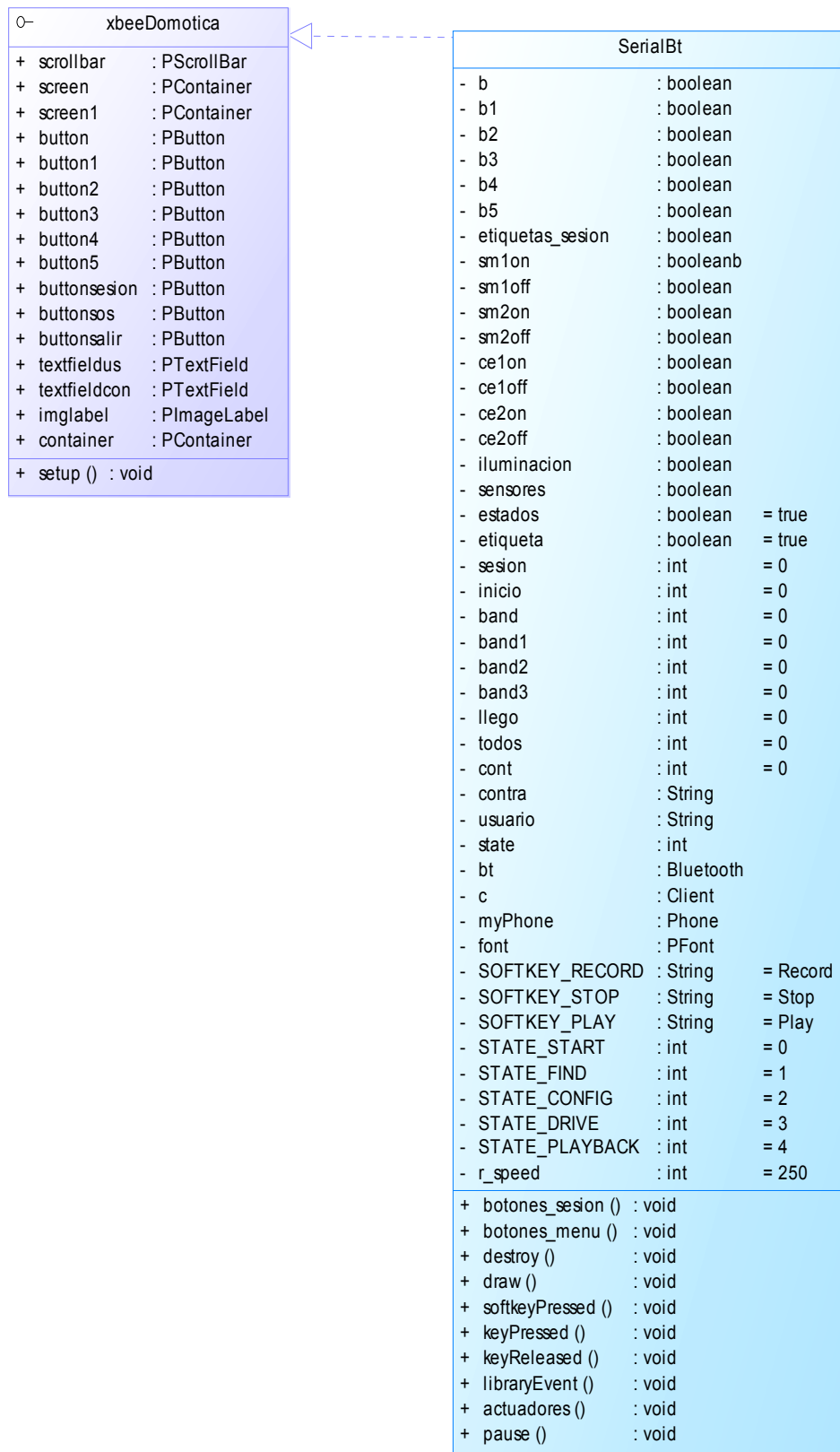


Figura 56: Diagrama de Clases Aplicación Bluetooth

### 4.5.3. SERVIDOR PRINCIPAL

Para la programación del servidor Principal se usa el desarrollador Netbeans IDE 6.5.1, Java Web, Visual Web JavaServerFaces.

El servidor principal escucha tres puertos COM como muestra la Figura 57.

Uno para el modem GSM enviar y recibir SMS, otro para la interfaz bluetooth establece conexión con un dispositivo bluetooth móvil y el de la interfaz Zigbee establece conexión con todos los módulos de comunicaciones y los dispositivos finales.

Están conectados al servidor de manera USB simulando puertos COM virtuales.

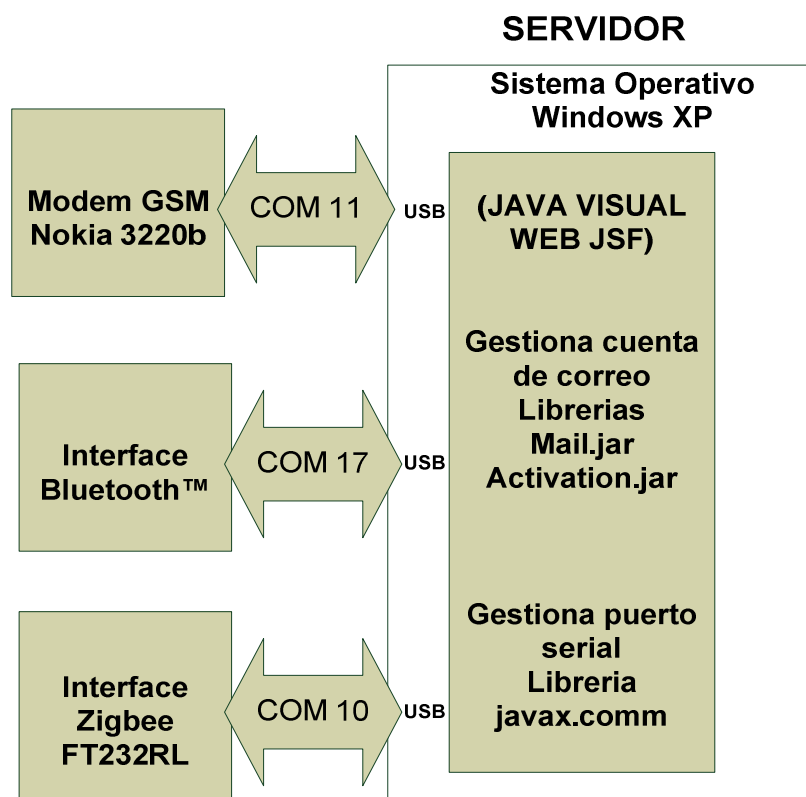


Figura 57: Diagrama de Bloques puertos que usa el servidor

Para abrir un puerto COM desde Java se necesita la librería javax.comm de esta manera es posible establecer comunicación con cada puerto individualmente.

#### **4.5.3.1. Módulos servidor principal**

El servidor principal se divide en 6 módulos.

- Módulo Servidor SMS
- Módulo Servidor Bluetooth
- Módulo Servidor Zigbee
- Módulo Servidor Correo
- Módulo Servidor Cámara IP
- Módulo Servidor Base de datos

##### **4.5.3.1.1. Módulo Servidor SMS**

Este módulo permite la comunicación vía SMS con teléfonos celulares, se crea un hilo que este escuchando el puerto COM determinado con la ayuda de la librería javax.comm, en busca de algún SMS nuevo, si entiende alguno realiza la acción determinada si no sigue escuchando.

En la siguiente figura, se muestra el diagrama de bloques del módulo servidor SMS

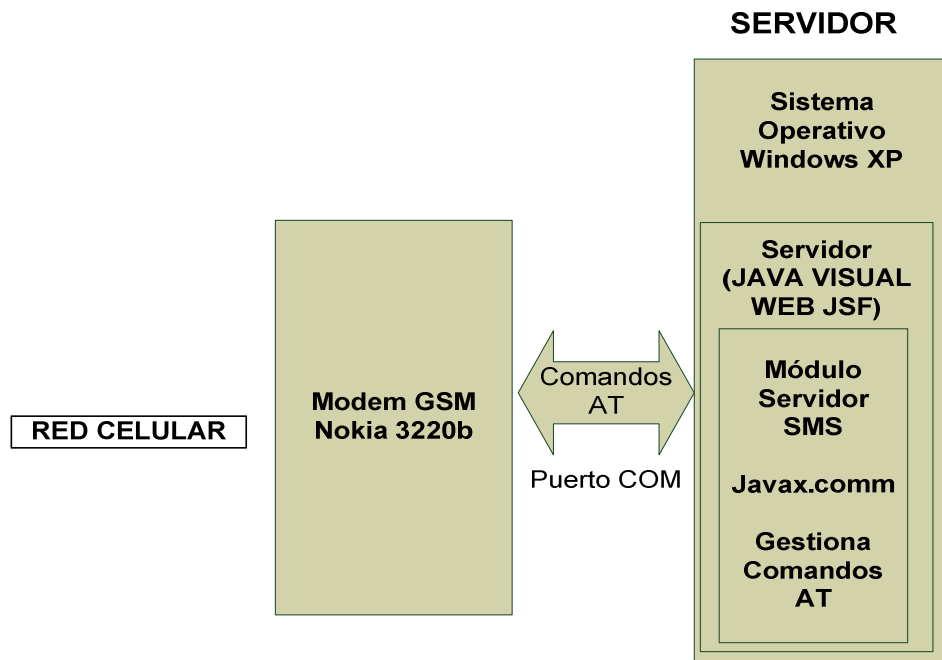


Figura 58: Diagrama de Bloques módulo servidor SMS

Si llega un mensaje de texto SMS el servidor primero compara el número de remitente con su base de datos, si el número existe procede a comparar el mensaje recibido y realiza la acción específica.

Este módulo es capaz de enviar SMS, llamando a un método enviar SMS cuando sea requerido, este método detiene el proceso de leer mensajes nuevos y usa el puerto para enviar, luego reinicia el proceso de lectura de mensajes.

El usuario administrador decide bajo qué circunstancias el módulo servidor SMS envía mensajes de notificación.

Se puede monitorear los siguientes grupos de dispositivos finales:

Iluminación

Detectores de movimiento

Detector de Humo

Detector de GAS

Sensores Magnéticos



El usuario administrador selecciona uno o varios dispositivos finales a monitorear y cuando suceda algún evento en el dispositivo final se le notifica al número de celular lo sucedido, por medio de un SMS de esta manera es posible estar informado en todo momento y lugar lo que sucede en el hogar.

#### 4.5.3.1.2. Módulo Servidor Bluetooth

Este módulo permite la comunicación con un puerto COM virtual con la ayuda de la librería javax.comm y este a su vez con un dispositivo remoto que tenga iniciada la aplicación Bluetooth como muestra la siguiente figura.

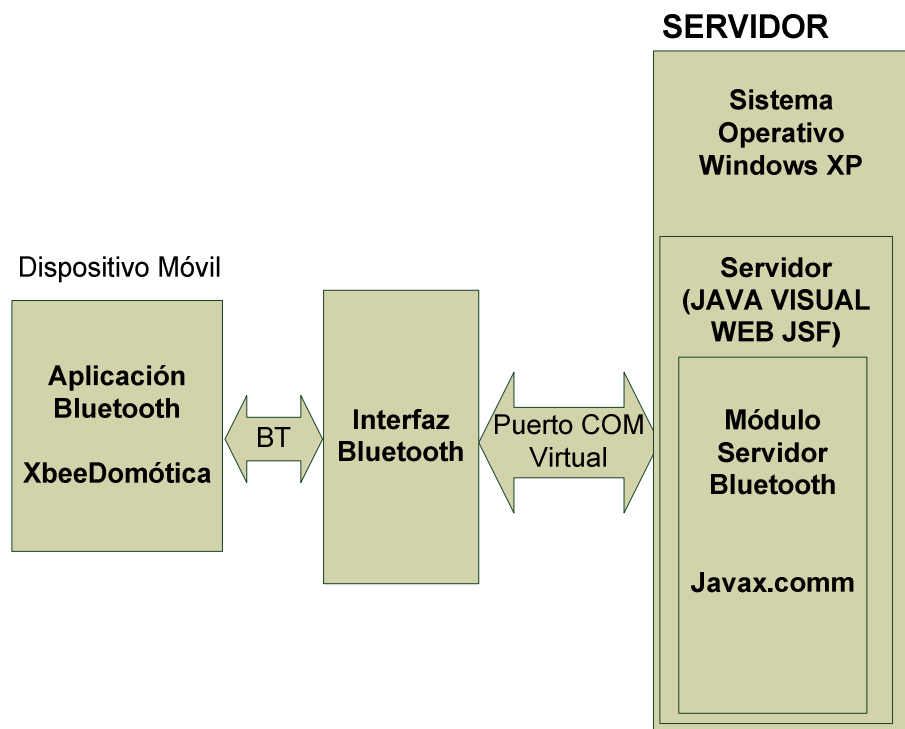


Figura 59: Diagrama de Bloques módulo servidor Bluetooth

El módulo servidor Bluetooth responde a peticiones realizadas por el cliente que usa la aplicación Bluetooth XbeeDomótica, todo se realiza de manera serial es transparente el tipo de medio que se

usa, por lo tanto, es similar al módulo SMS, se necesita un hilo que este escuchando peticiones nuevas en el puerto las compara y realiza la acción específica.

De igual manera envía datos por este puerto siempre y cuando sea solicitado por el cliente previamente se podría decir que es no orientado a conexión.

#### 4.5.3.1.3. Módulo Servidor Zigbee

Este módulo es el más importante de todos aquí se controlan los diferentes módulos de comunicaciones que contienen a los dispositivos finales.

Consiste en un hilo que está escuchando el puerto COM con la ayuda de la librería javax.comm en busca de algún mensaje nuevo, si llega uno compara y realiza la acción específica.

En la siguiente figura, se muestra el diagrama de bloques del módulo servidor Zigbee.

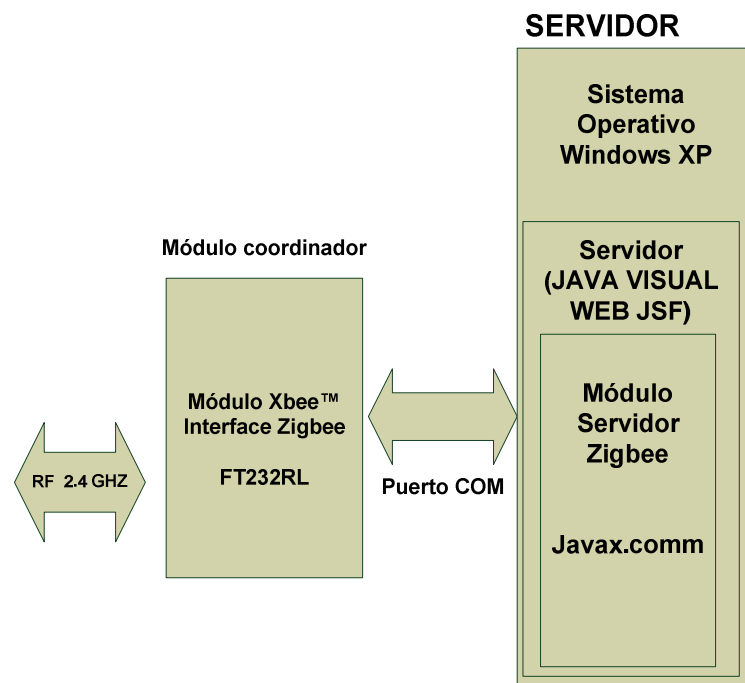


Figura 60: Diagrama de Bloques módulo servidor Zigbee

El módulo servidor Zigbee no solo responde a peticiones de los clientes Módulo Xbee 1, Módulo Xbee 2, Módulo Xbee 3 y Módulo GLCD, sino que también realiza peticiones a estos módulos con el fin de controlar y monitorear los dispositivos finales sensores y actuadores.

#### **4.5.3.1.4. Módulo Servidor Correo**

Este módulo usa las librerías Mail.jar y Activation.jar para gestionar una cuenta de correo electrónico en GMAIL y enviar notificaciones vía mail.

Se trata de una clase correo electrónico que gestiona la cuenta GMAIL usando las librerías antes descritas, inicia sesión proporcionando la dirección de correo y contraseña, es posible enviar un mail con dos copias a direcciones distintas, es decir, 3 cuentas de correo electrónico pueden recibir las notificaciones de monitoreo.

En la siguiente figura, se muestra el diagrama de bloques del módulo servidor Correo.

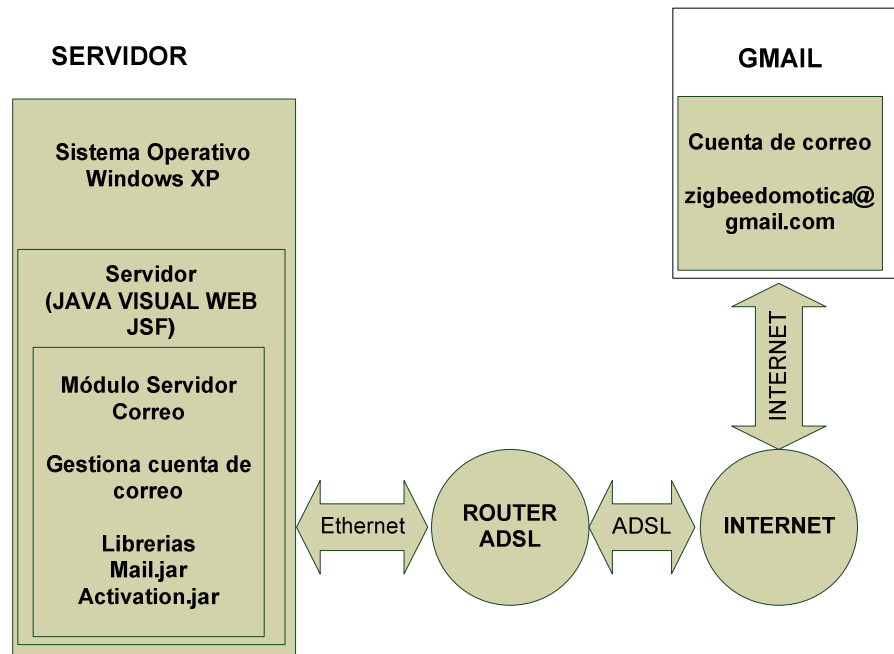


Figura 61: Diagrama de Bloques módulo servidor Correo

El usuario administrador decide bajo qué eventos el módulo servidor Correo envía mensajes de notificación.

Se puede monitorear los siguientes grupos de dispositivos finales:

Iluminación

Detectores de movimiento

Detector de Humo

Detector de GAS

Sensores Magnéticos

El usuario administrador selecciona uno o varios dispositivos finales a monitorear y cuando suceda algún evento en el dispositivo final, se le notifica a las cuentas de correo electrónico lo sucedido de esta manera es posible estar informado en todo momento y lugar lo que sucede en el hogar.

#### 4.5.3.1.5. Módulo Servidor Cámara IP

Este módulo consiste en mostrar la imagen de una cámara IP conectada a la misma red ,en el servidor principal usando código JSP embebido y un applet, en la siguiente figura, se muestra el diagrama de bloques del módulo servidor Cámara IP.

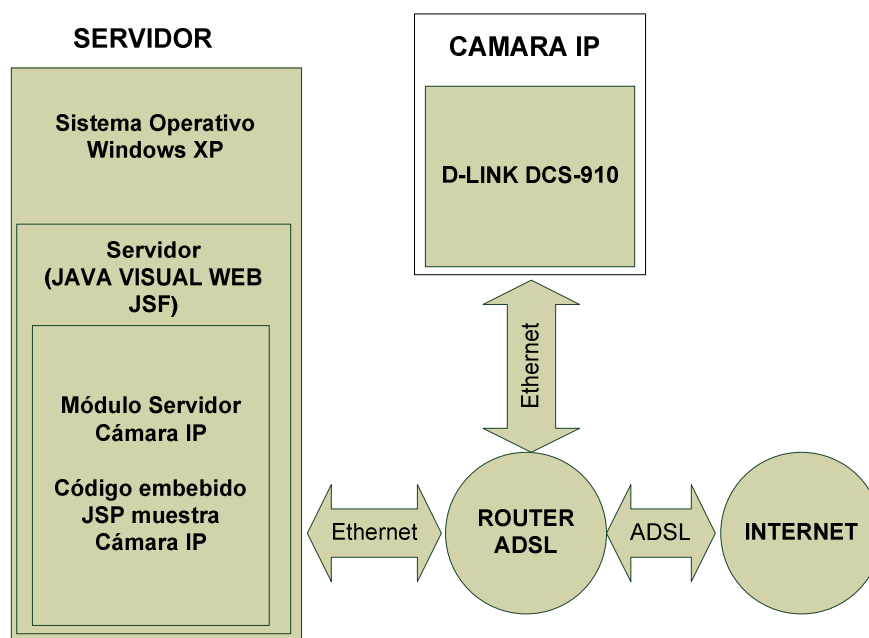


Figura 62: Diagrama de Bloques módulo servidor Cámara IP

#### 4.5.3.1.6. Módulo Servidor Base de datos

Este módulo consiste en la gestión de las tablas historial de la base de datos, historial general, historial Xbee e historial modos. Se lleva registro de cada una de los eventos que sucedan en el hogar.

##### Historial general

El usuario administrador decide bajo qué eventos el módulo servidor Base de Datos guarda en el historial general los eventos sucedidos en:

Iluminación  
Detectores de movimiento  
Detector de Humo  
Detector de GAS  
Sensores Magnéticos  
Cerraduras  
Sensor Fotovoltaico  
Bocina  
Cortina

El usuario administrador selecciona uno o varios dispositivos finales a monitorear, y cuando suceda algún evento en el dispositivo final esta información se almacena en la tabla historial de la base de datos con la fecha y hora, de esta manera es posible recuperar esta información cuando se necesite para saber que sucedió y cuando sucedió.

### **Historial Xbee**

En este historial el sistema almacena la información del estado de los módulos de comunicaciones Xbee 1, Xbee2 y Xbee 3, el estado está relacionado con el fluido eléctrico, cuando falla en alguno de estos módulos se guarda en la base de datos la hora y la fecha en que sucedió este evento y si el fluido eléctrico es restablecido de igual manera.

Es posible llevar un registro en la base de datos de cuando el fluido eléctrico falla o esta correcto y es posible recuperar esta información cuando se necesite.

### **Historial modos**

En este historial el sistema almacena la información del estado de los modos que tiene este sistema.

Modo Simulador de presencia

Modo Ahorro de energía

Modo alarma sonora

Se lleva registro de cuando fueron activados o desactivados estos modos fecha, hora y el usuario que los desactivo.

#### **4.5.3.2. Diagrama de navegación**

El diagrama de navegación permite tener una idea clara y gráfica de los caminos o rutas por los cuales el usuario debe transitar, este tipo de diagramas son de gran ayuda tanto para programadores como para los usuarios del sistema, ayuda a tener una idea clara de los caminos a seguir para llegar a su objetivo.

**Nota:** El diagrama de navegación se observa en el Anexo 14.

#### **4.5.3.3. Estructura de Páginas**

Esta etapa considera la generación de dibujos sólo lineales que describen los componentes de cada una de las pantallas del sitio, con el objetivo de verificar la ubicación de cada uno de ellos.

De esta manera será posible que diseñadores y desarrolladores tengan un documento concreto de trabajo, a través del cual resuelvan todas las dudas de los elementos que componen esta página.

**Nota:** Estructura de las páginas se observa en el Anexo 15.

#### 4.5.3.4. Bocetos de diseño

Esta etapa consiste en la generación de dibujos digitales acabados de la forma que tendrán las páginas principales del sitio que se desarrolla, considerando como tales la Portada y Página de despliegue de contenidos, en las siguientes figuras, se muestran los bocetos de diseño del Sistema.

El boceto muestra una interfaz de usuario para el sistema 'Xbee Domótica'. En la parte superior, un encabezado amarillo contiene el título 'Xbee Domótica'. Debajo, se encuentran tres campos de entrada etiquetados como 'Usuario', 'Contraseña' y 'Tipo', cada uno con un asterisco rojo indicando que son obligatorios. El campo 'Tipo' tiene un menú desplegable que muestra 'abc'. Debajo de los campos hay un botón rectangular etiquetado 'Iniciar'. En la parte inferior, una barra de pie contiene el texto 'Universidad Politécnica Salesiana Quito - UPSQ' a la izquierda, los nombres 'Miguel Ángel Cedeño' y 'Carlos Eraz' en el centro, y el logo 'Xbee Domótica' a la derecha.

Figura 63: Boceto de diseño Ingreso de usuario

El boceto muestra la interfaz de la portada principal del sistema 'Xbee Domótica'. El encabezado amarillo superior contiene el título 'Xbee Domótica'. Debajo, una barra blanca contiene el texto 'Bienvenido : ' seguido de un campo de texto con el valor 'Label' y un botón 'Cerrar Sesión' a la derecha. El contenido principal está dividido en dos secciones. A la izquierda, un menú vertical con el título 'Menú' contiene varias opciones: 'Monitoreo' (con un icono de reloj), 'Control' (con un icono de interruptor), 'Notificaciones' (con un icono de campana) y 'Gestión' (con un icono de usuario). Cada opción tiene un submenú desplegado. A la derecha, el texto 'BIENVENIDOS' y 'CONTROL DEL HOGAR DIGITAL' están centrados. Debajo de este texto, un recuadro con un fondo gris claro contiene el texto: 'Diseño, implementación y monitoreo de una red Mesh/Zigbee aplicado a la domótica, gestionada mediante SMS (Short Message Service), servidor web, correo electrónico, aplicación bluetooth para dispositivos móviles y control inalámbrico local.' En la parte inferior, una barra de pie contiene el texto 'Universidad Politécnica Salesiana Quito - UPSQ' a la izquierda, los nombres 'Miguel Ángel Cedeño' y 'Carlos Eraz' en el centro, y el logo 'Xbee Domótica' a la derecha.

Figura 64: Boceto de diseño Portada Principal



#### 4.5.3.5. Mapa permanente del sitio

Se refiere al proceso de crear un árbol de contenido en el que se muestre de manera práctica cuántas secciones tendrá el sitio en desarrollo y cuántos niveles habrá dentro de cada uno.

Cuando se usa la idea de crear un árbol, se refiere exactamente a generar un diagrama que cuente con un tronco, ramas y hojas, para mostrar las zonas principales, secundarias y contenidos finales que se irán incorporando.

Se crea un árbol funcional por cada tipo de usuario existente en el sistema, en este caso un Administrador y un Invitado.

Cada Árbol representa zonas a las que tiene acceso cada tipo de usuario, en las siguientes figuras, se muestra el árbol funcional de cada tipo.

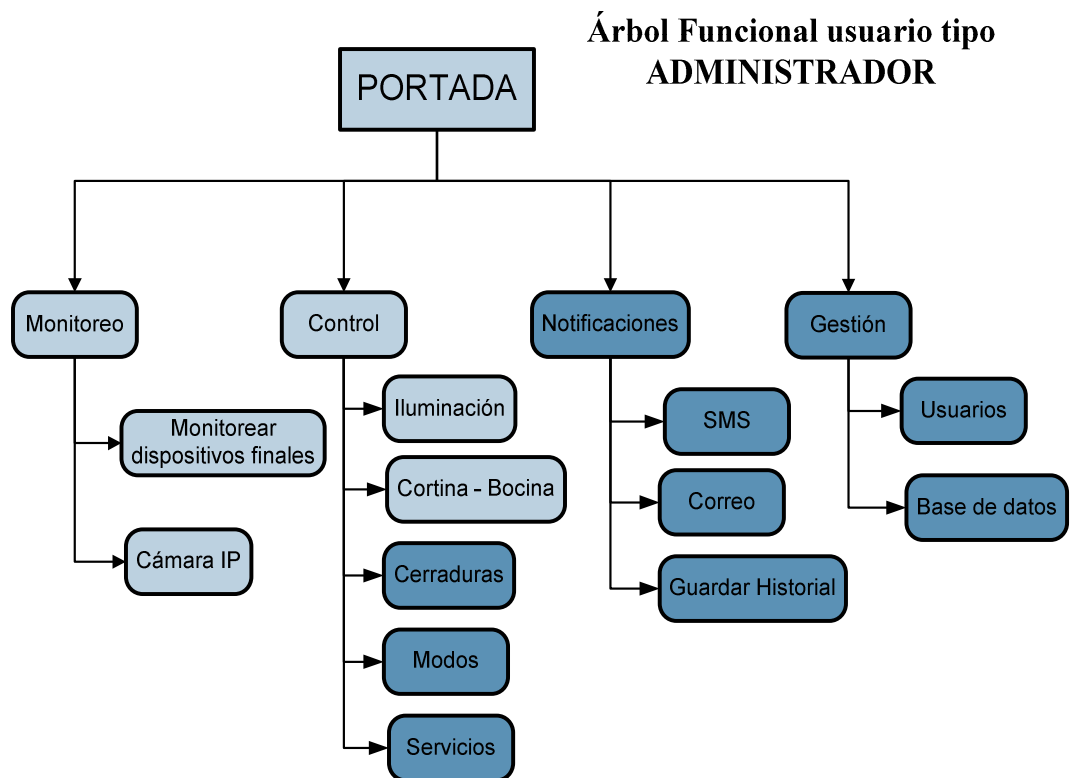


Figura 65: Árbol funcional usuario tipo ADMINISTRADOR

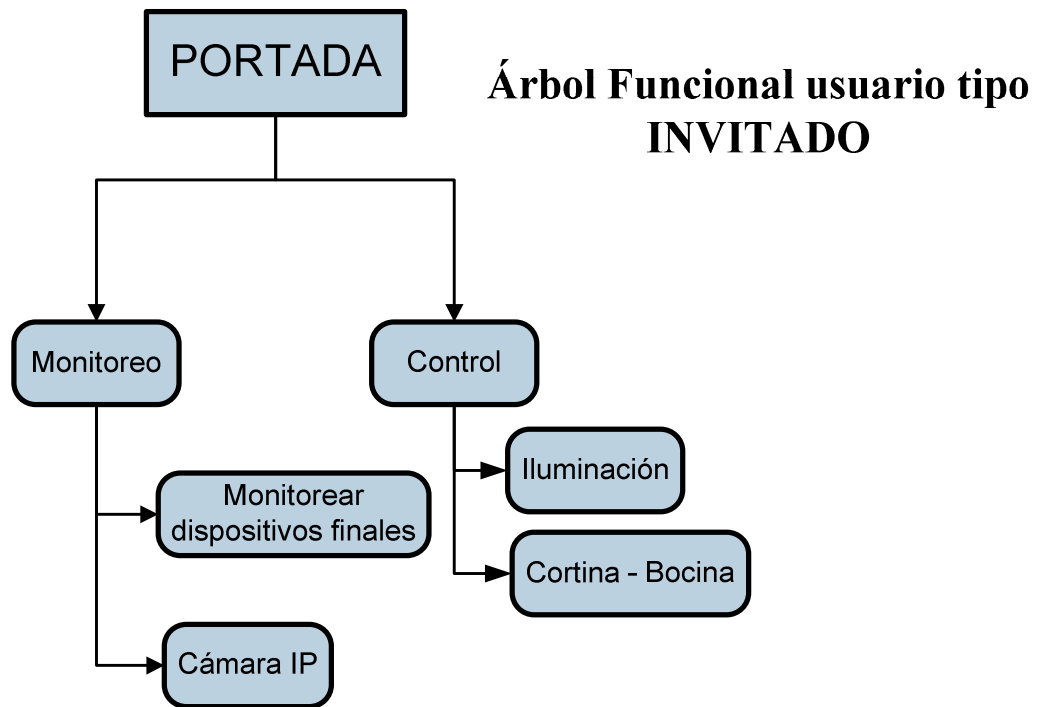


Figura 66: Árbol funcional usuario tipo INVITADO

#### 4.5.3.6. Diagrama de Flujo

**Nota:** El diagrama de flujo se observa en el Anexo 16.

#### 4.5.3.7. Casos de uso

**Nota:** El diagrama de casos de uso se observa en el Anexo 17.

#### 4.5.3.8. Diagrama de clases

**Nota:** El diagrama de clases se observa en el Anexo 20.

#### 4.5.3.9. Seguridades

Es importante que el sistema tenga tolerancia a fallos y redoblaje de seguridades, no se puede tener un sistema vulnerable que controle la

seguridad de un hogar especialmente las partes sensibles como son apertura de cerraduras.

#### **4.5.3.9.1. Control de Usuarios**

El sistema realiza el control de usuarios al ingresar al sitio web, pide el nombre de usuario, contraseña y tipo de usuario con el fin de restringir el acceso a personas ajenas al sistema.

#### **4.5.3.9.2. Respaldos Base de datos**

Tener respaldos de la Base de datos es una estrategia de seguridad muy buena, si un intruso logra ingresar al sistema borra o modifica la base de datos se puede recuperar restaurando la copia de seguridad anterior a esa modificación.

Los respaldos serán periódicos cada mes y manuales cuando el usuario lo considere necesario.

#### **4.5.3.9.3. Encriptación Base de datos**

Tener una base de datos encriptada es la mejor opción para solucionar los problemas de seguridad que podría tener MySQL, en este sistema se encriptará la contraseña de todos los usuarios con el fin de que un intruso que ingrese a la base de datos, de modo solo lectura no pueda entender lo que se almacena en el campo contraseña.

# CAPÍTULO 5

## IMPLEMENTACIÓN

Este capítulo hace referencia a la implementación del sistema, configuraciones, desarrollo de hardware y generación de código de software.

### 5.1. CONFIGURACIONES

Red Zigbee Mesh con módulos Xbee series 2 y Bluetooth (Emparejamiento, Selección de servicios).

#### 5.1.1. ZIGBEE

##### Configuración de módulos Xbee para crear la red Mesh

Instalar el XCTU para gestionar los módulos Xbee de una manera sencilla y gráfica.

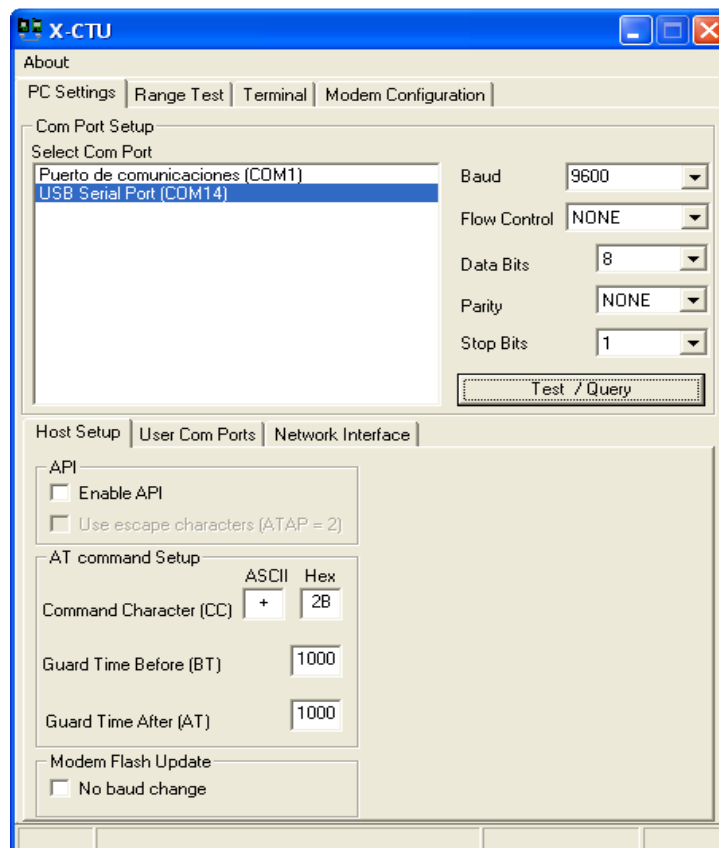


Figura 67: Pantalla principal del XCTU

Clic en Test / Query, para verificar que el módulo Xbee esté conectado a la interfaz USB ubicada en el puerto 14 en este caso

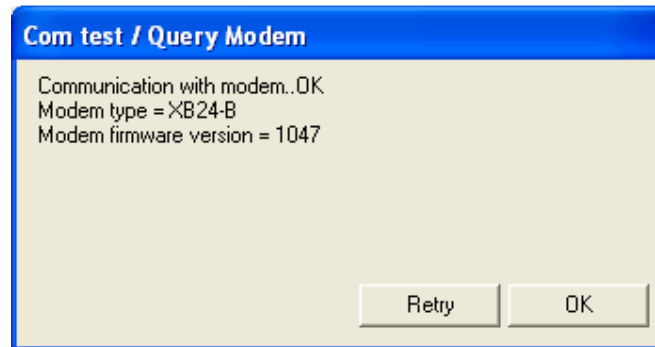


Figura 68: Verifica si el modem Xbee está conectado

## Configuración del Coordinador

En la pestaña **Modem Configuration** se configura el módulo Xbee.

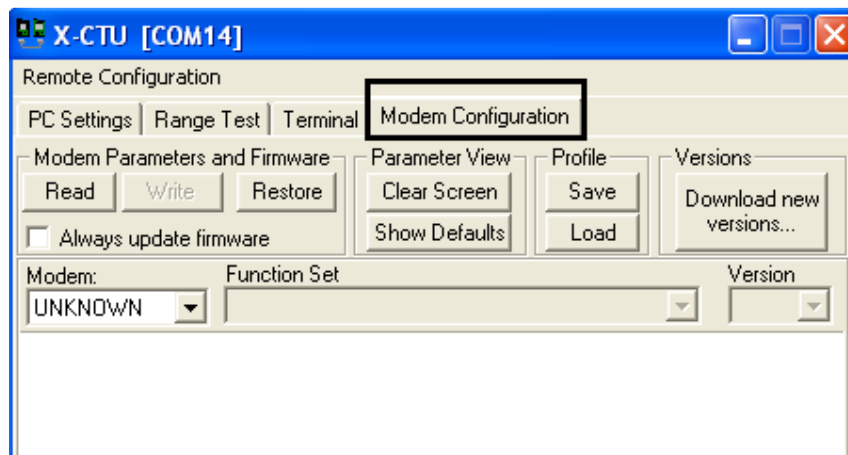


Figura 69: Muestra pestaña Modem Configuration

Se verifica versiones nuevas dando clic en el botón “Download new versions” por medio de la web.

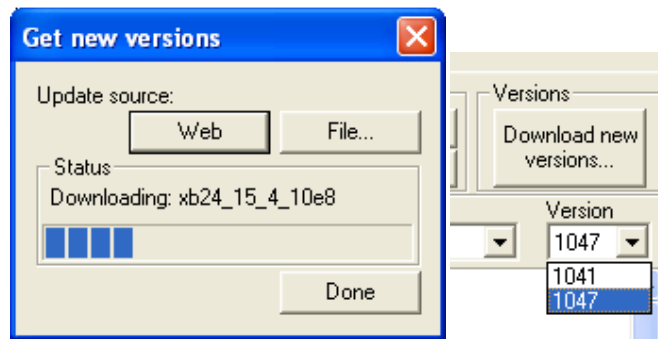


Figura 70: Obteniendo nuevas versiones de software Xbee

Con ayuda del XCTU Modem Configuration, configurar los siguientes valores en el Xbee coordinador.

Para realizar una red Mesh se utiliza la función “ZNET 2.5 COORDINATOR AT”.

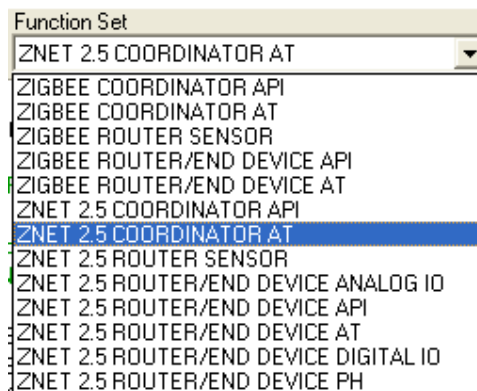


Figura 71: Configuración de funciones Xbee

Clic en “write” para guardar los cambios realizados.

Se configura el coordinador bajo estos parámetros:

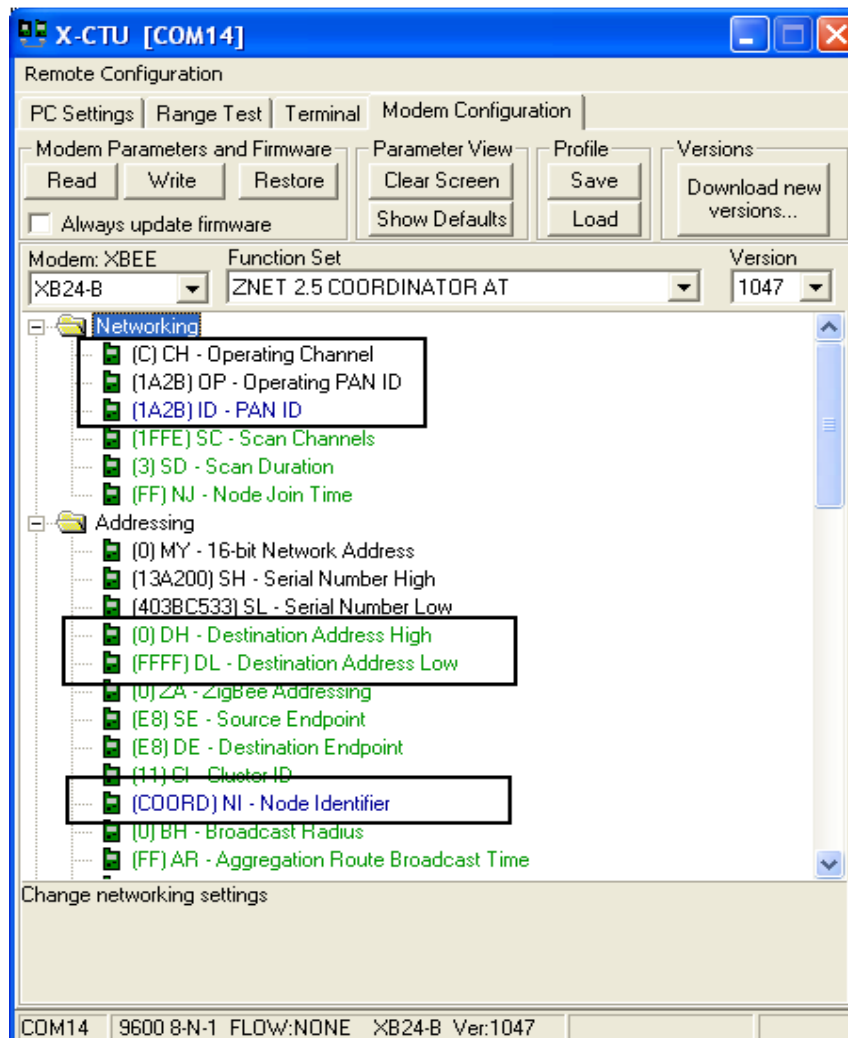


Figura 72: Parámetros para configurar el coordinador

El canal de comunicaciones operativo se selecciona automáticamente haciendo un análisis de la carga de cada canal en el ambiente y selecciona el que tenga menos energía.

Por seguridad es recomendable cambiar el PAN ID que viene por defecto (234).

SH Y SL representa el serial alto y bajo del Xbee, siendo el serial bajo único e identificador de cualquier dispositivo se podría comparar con la dirección MAC de una tarjeta de red.

Los campos DH Y DL son para identificar los nodos con los cuales el coordinador va a establecer comunicación, el FFFF de DL es un

broadcast esto quiere decir que se va a comunicar con todos los nodos queestén en el mismo PAN ID y comparten la misma clave AES.

Para identificar el Xbee como coordinador en Node identifier se coloca COORD.

Estos pasos son posibles realizarlos por medio de comandos AT con la ayuda del Hyperterminal sin la necesidad de XCTU, para facilidad se usa este software.

```
+++ OK          // Pone al Xbee en Modo comandos AT
ATID 1A2B       // Cambia el PAN ID
OK
ATNICOORD       // Nombre del Nodo
OK
ATWR            // Guarda cambios hechos WRITE
OK
ATCN            // Sale del modo comando AT
OK
```

### Configuración del Router o dispositivo final

Para realizar una red Mesh se utiliza la función “ZNET 2.5 ROUTER/END DEVICE AT” para configurar el Xbee Router o dispositivo final.

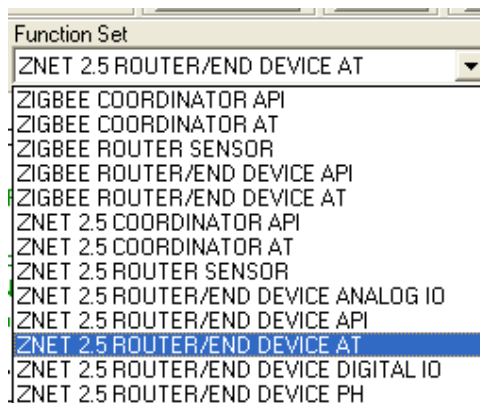


Figura 73: Configuración de funciones Xbee



Se configura el Router bajo estos parámetros:

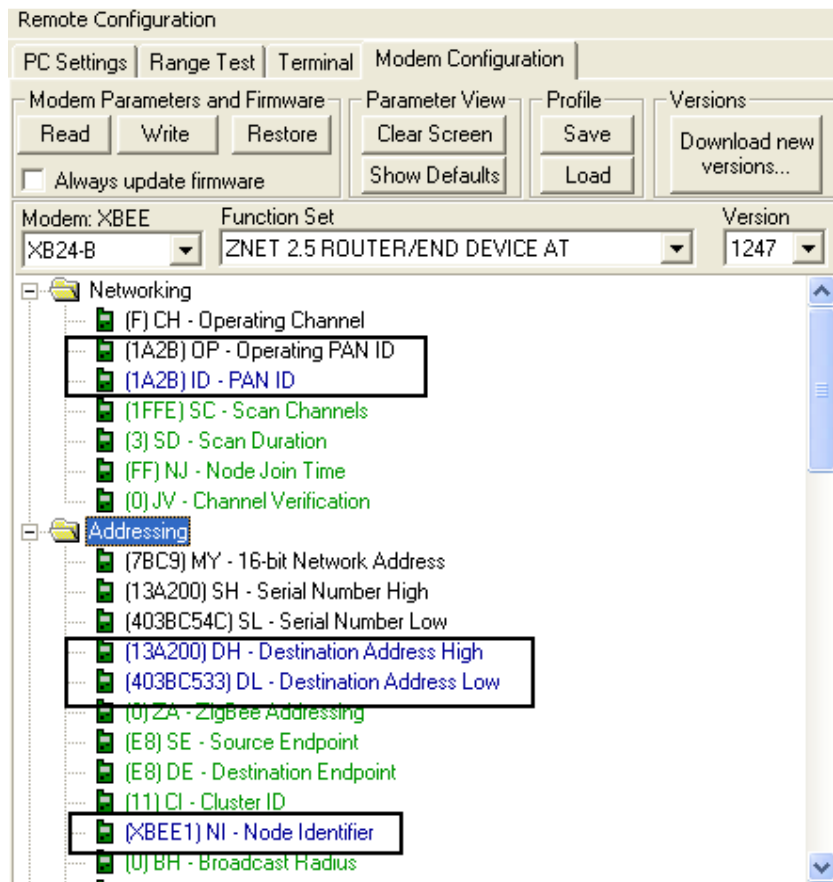


Figura 74: Parámetros para configurar el Router o end device

Note que los parámetros DH Y DL corresponden a SH Y SL del coordinador, con esta configuración el Router se comunicara con el coordinador asignado.

En este caso copiar los parámetros se hace un poco difícil por lo que el uso de comandos AT facilita mucho este trabajo.

+++OK

ATNI XBEE1 // Nombre del Router

OK

ATND // Comando para descubrir los nodos en el PAN ID

0000

0013A200

403BC533

COORD

FFFE

00

00

C105

101E

ATDN COORD // Relaciona el coordinador con el Router

OK

ATWR

OK

ATCN

OK

Para verificar que los nodos Router y coordinador sí establecen comunicación entre sí, usando el terminal del XCTU se digita cualquier caracter, si todo está correctamente configurado en el terminal del coordinador se visualizará lo que el Router escribe y viceversa.

En la siguiente figura, se muestra la prueba de conexión entre dos dispositivos Xbee usando el Hyperterminal de dos computadores.

## PRUEBA DE CONEXIÓN XBEE ENTRE EL ROUTER Y EL COORDINADOR

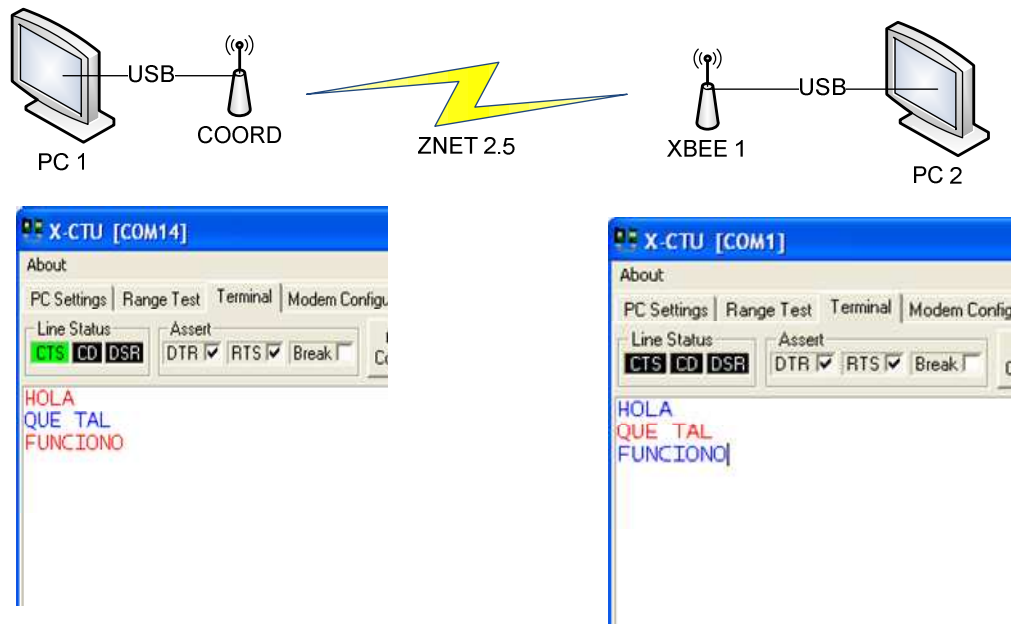


Figura 75: Prueba de conexión básica entre el Router y el coordinador Xbee

Para configurar los Xbee1, Xbee2, Xbee3 y Xbee4 (GLCD) se siguen los pasos de configuración del Router o end device porque en la red que se propone solo existe un coordinador.

### Configuración de seguridad en los módulos Xbee serie 2

La configuración de seguridad de la red Zigbee Mesh es muy importante para garantizar el acceso solo de usuarios autorizados y de esta manera prevenir ataques de individuos infiltrados en nuestra red que podrían ocasionar graves daños en el sistema.

Activar el modo encriptación y proporcionar una clave AES, para que los datos que viajen en esta red sean entendidos solo por los nodos que

estén en el mismo canal , compartan el mismo PANID y tengan la clave AES.

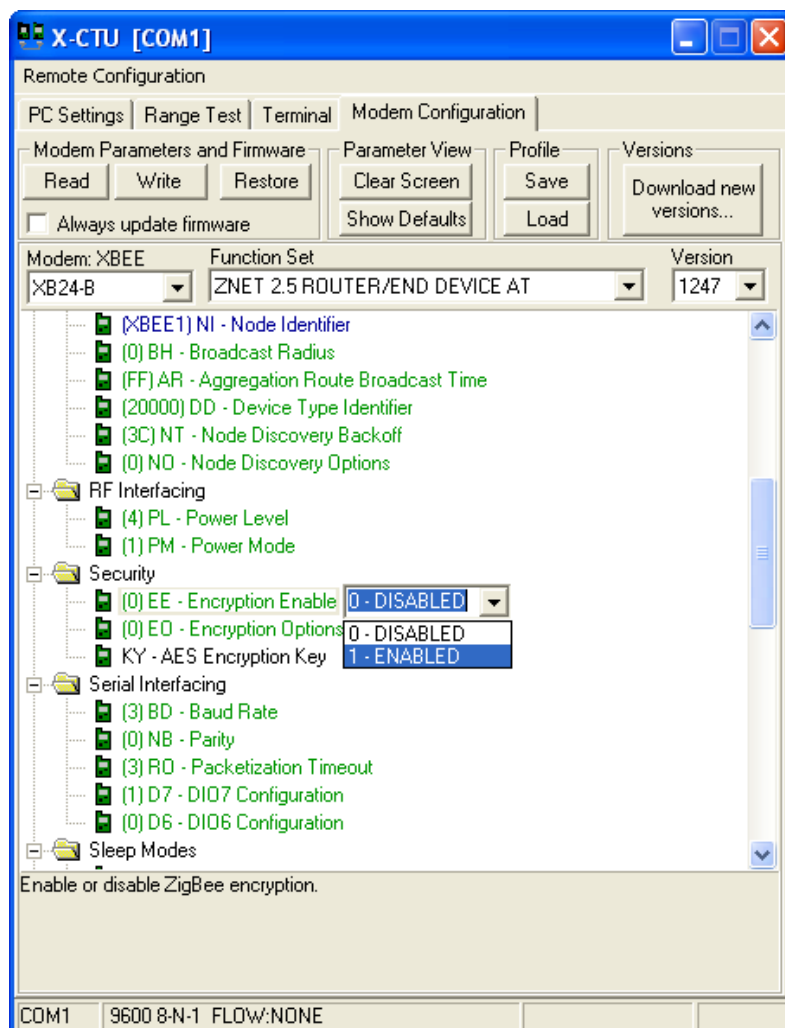


Figura 76: Parámetros de configuración de seguridad inalámbrica ZNET

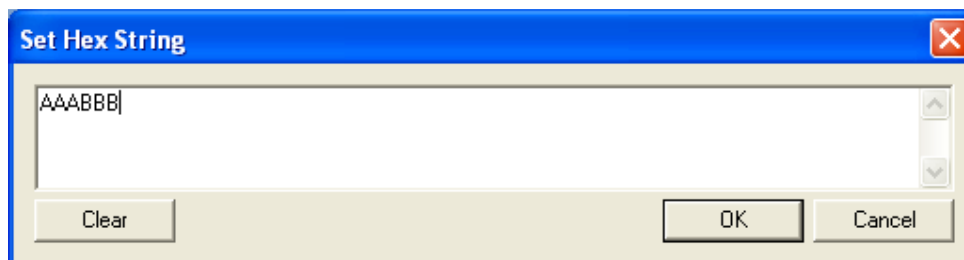


Figura 77: Clave AES

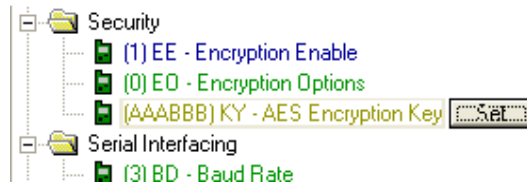


Figura 78: Parámetros de seguridad configurados

Clic en write para que los cambios se guarden correctamente.

Configuración de seguridad por medio de comandos AT.

+++OK

ATEE 1 // Activa la encriptación

OK

ATKY AAABBB // Clave de encriptación AES

OK

ATWR

OK

ATCN

OK

### 5.1.2. BLUETOOTH

Para establecer conexión entre dispositivos bluetooth es necesario emparejarlos, para proporcionar una clave y la red sea segura. Se requiere emparejar el dispositivo cliente en este caso el teléfono celular con el dispositivo servidor una computadora con Bluetooth USB.

#### Emparejar dispositivos bluetooth

En el servidor se utiliza el asistente bluetooth para agregar dispositivos nuevos, clic derecho sobre el icono bluetooth ubicado en la barra inicio, luego clic en Agregar dispositivo Bluetooth.



**Figura 79: Asistente agregar dispositivo Bluetooth**

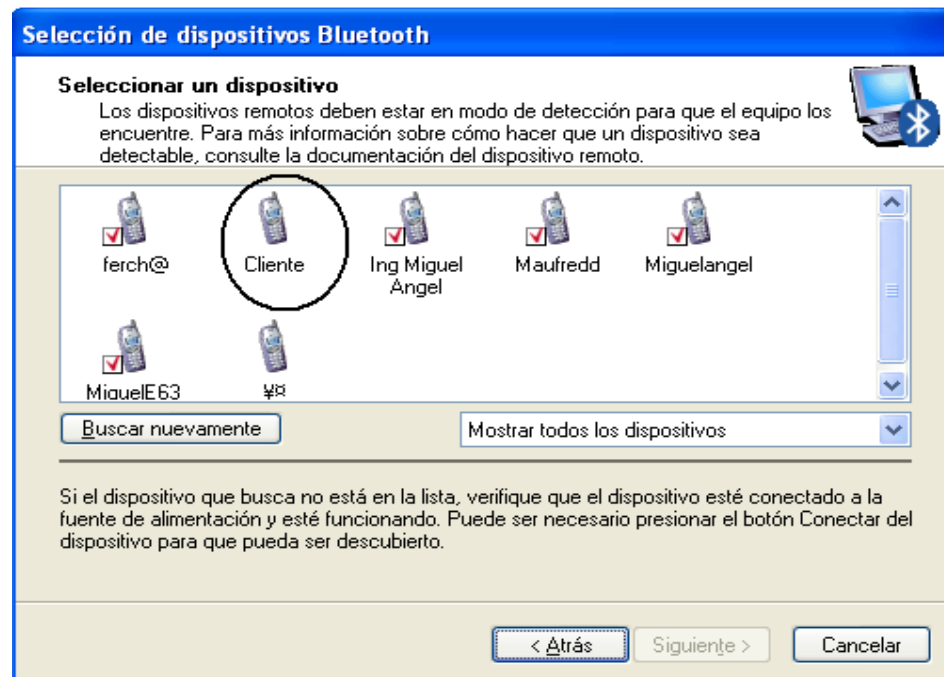
Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122



**Figura 80: Configuración Bluetooth**

Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122

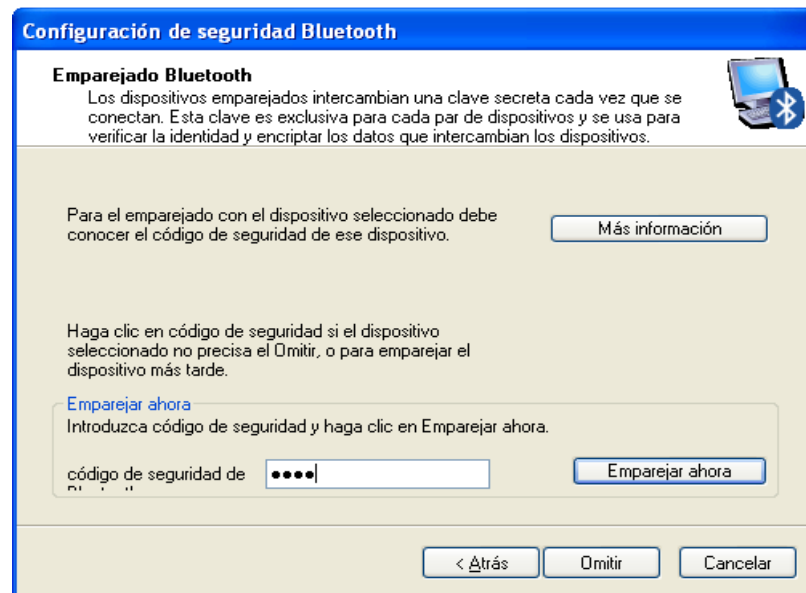
Seleccionar el dispositivo bluetooth cliente a emparejar, luego clic en siguiente, si no aparece en la lista compruebe la visibilidad del cliente y clic en buscar nuevamente.



**Figura 81: Selección de dispositivo Bluetooth**  
Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122

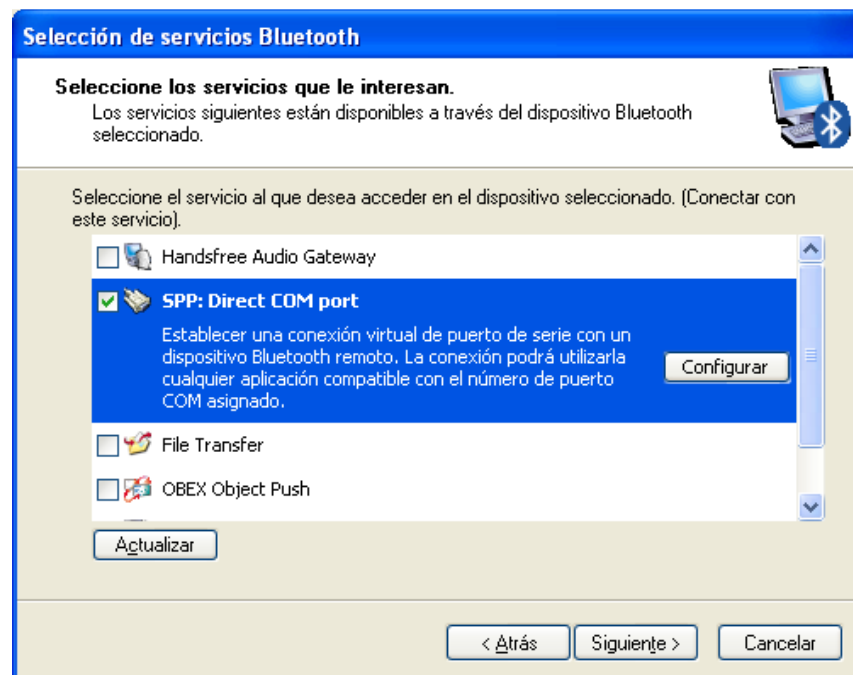
La configuración de seguridad es clave para emparejar dispositivos bluetooth, se debe colocar la misma clave del servidor en el cliente, una vez emparejados intercambian esta clave cada vez que se conecten.

Colocar el código de seguridad que se crea conveniente y clic en emparejar ahora, de inmediato en el dispositivo cliente (celular) pedirá que se ingrese este código.



**Figura 82: Configuración de seguridad Bluetooth**  
Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122

En la selección de servicios Bluetooth activar SSP: Direct COM port, Este servicio permite establecer una conexión virtual de puerto serie con dispositivo bluetooth remoto, la conexión podrá usarla cualquier aplicación compatible con el número de puerto COM asignado.



**Figura 83: Selección de servicios Bluetooth**  
Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122



Y finaliza el asistente, se tienen los dispositivos bluetooth emparejados y levantado el servicio de puertos COM virtuales en el servidor.



**Figura 84: Finalización del asistente de configuración Bluetooth**  
Fuente: Software Configuración Bluetooth dispositivo D-LINK DBT 122

## 5.2. Desarrollo de Hardware

Desarrollo del hardware de cada uno de los módulos y circuitos básicos que componen el sistema.

**Nota:** El costo de los componentes usados en la fabricación de los circuitos se observa en el Anexo 18.

### 5.2.1. CIRCUITOS BÁSICOS

Se denomina circuitos básicos al hardware que compone de forma complementaria a cada uno de los módulos del sistema.

### 5.2.1.1. Conexión módulo Zigbee – Microcontrolador

Para energizar un módulo Zigbee (Xbee series 2) se necesita un regulador de voltaje llamado LD33CV o LF33 la idea es reducir la alimentación de 5 vcc a 3.3 vcc.

La conexión entre un módulo Zigbee (Xbee series 2) y un microcontrolador se realiza directamente solo con la ayuda de 3 resistencias de 10 K $\Omega$  para disminuir el voltaje que emite el microcontrolador por su salida de datos serial, porque trabaja con 5 vcc y el módulo Zigbee con 3.3 vcc en la figura 85, se muestra la conexión de alimentación de voltaje del módulo Xbee y la comunicación serial con el PIC 18F452.

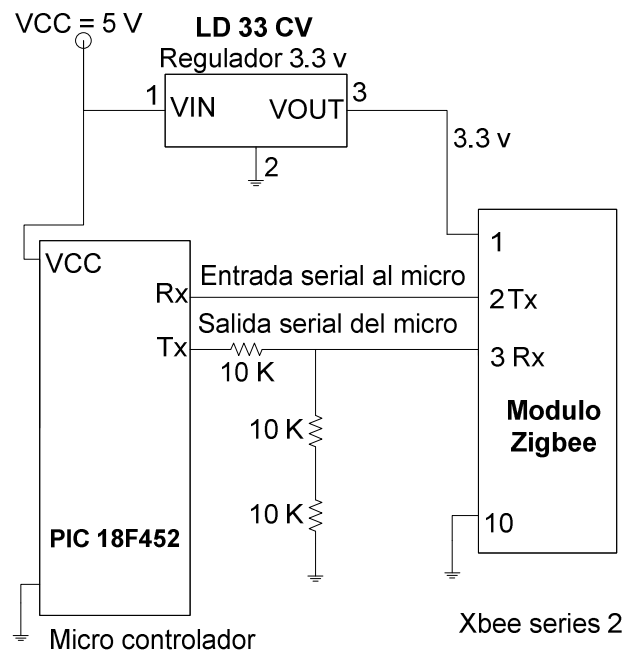


Figura 85: Conexión módulo Zigbee y microcontrolador

### 5.2.1.2. Multiplexor de voltaje

La función del Multiplexor de voltaje es la de proveer energía ininterrumpida por medio del suministro local y si falla usa una batería de 9 voltios, de esta manera el circuito principal no se ve afectado por la falta de suministro eléctrico y sigue funcionando de manera habitual, se garantiza disponibilidad, el cambio de suministro local a la batería de 9 voltios y viceversa es transparente para el circuito ocurre de manera automática. En la Figura 86, se muestra el diagrama circuital del multiplexor de voltaje, se usan dos diodos rectificadores.

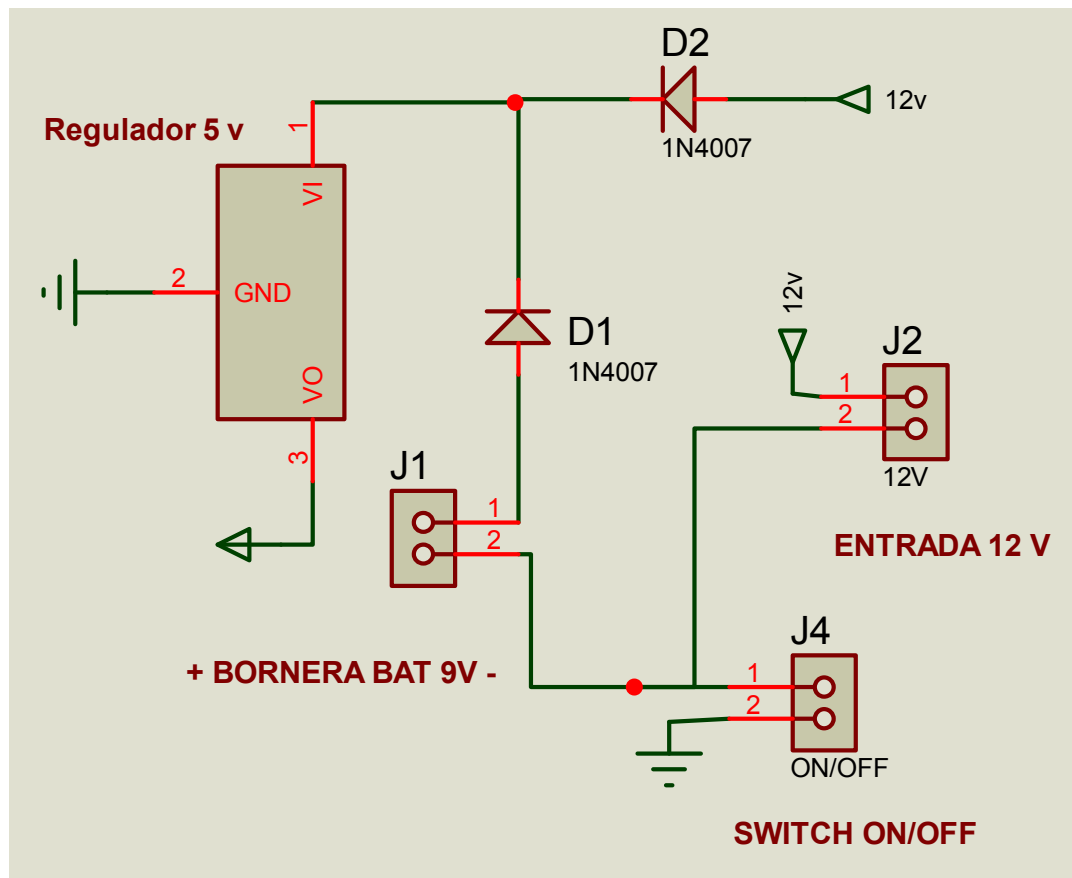


Figura 86: Diagrama circuital Multiplexor de voltaje

### 5.2.1.3. Detector de energía

El sistema necesita saber cuando sufrió un corte de fluido eléctrico, por lo que es necesario crear un detector de energía con el fin de establecer hora y fecha en que se sufrió esta falla, para garantizar la seguridad del sistema domótico y notificar al usuario lo que está sucediendo de manera inmediata, en la Figura 87, se muestra el diagrama circuital del detector de energía, funciona conjuntamente con el multiplexor de voltaje, cuando el fluido eléctrico sufre una falla inactiva el relé y envía una señal al microcontrolador indicando lo sucedido.

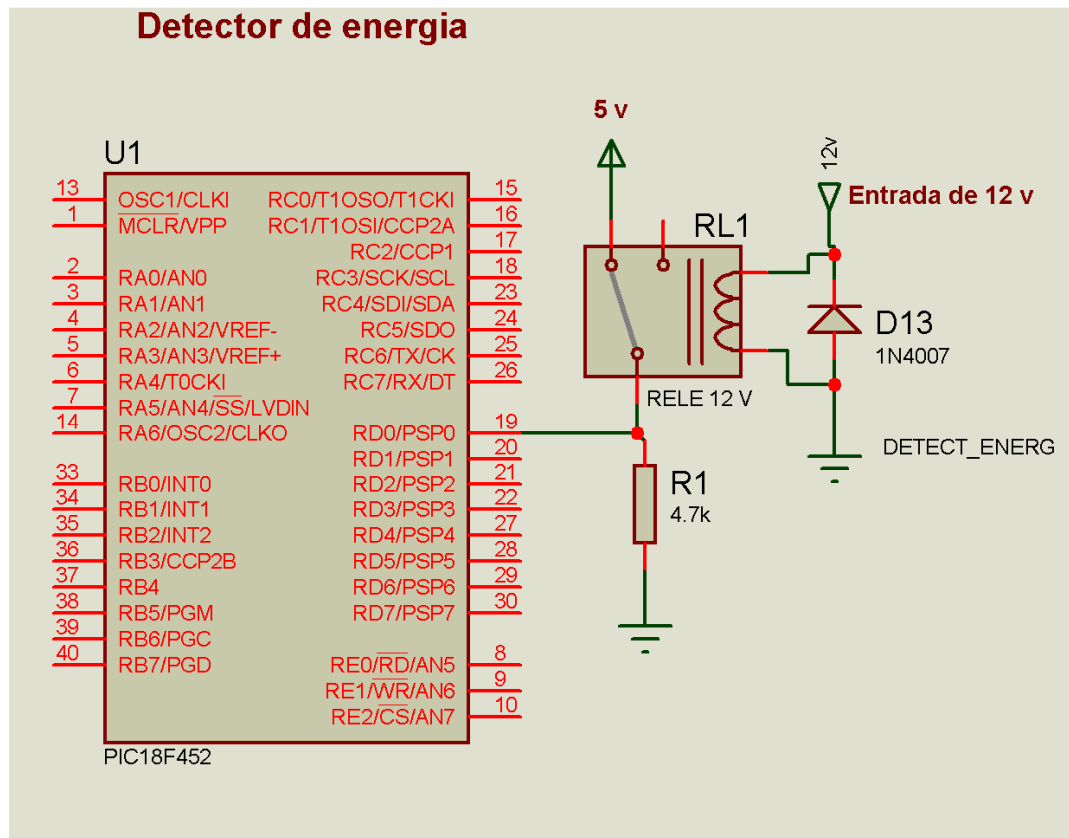


Figura 87: Diagrama circuital Detector de energía

#### 5.2.1.4. Detector de Luz

Este dispositivo envía al microcontrolador una señal analógica que varía dependiendo de la cantidad de luz presente en la habitación, la fotorresistencia incrementa o disminuye su resistividad ante la presencia o ausencia de luz, en la figura 88, se muestra el diagrama de simulación de una fotorresistencia envía una señal analógica que luego va a ser interpretada por el microcontrolador.

#### Materiales

- 1 Resistencia 10 k  $\Omega$
- 1 Fotorresistencia LDR
- 1 Bornera de 3 contactos
- 1 Baquelita de cobre de 4 x 4 cm

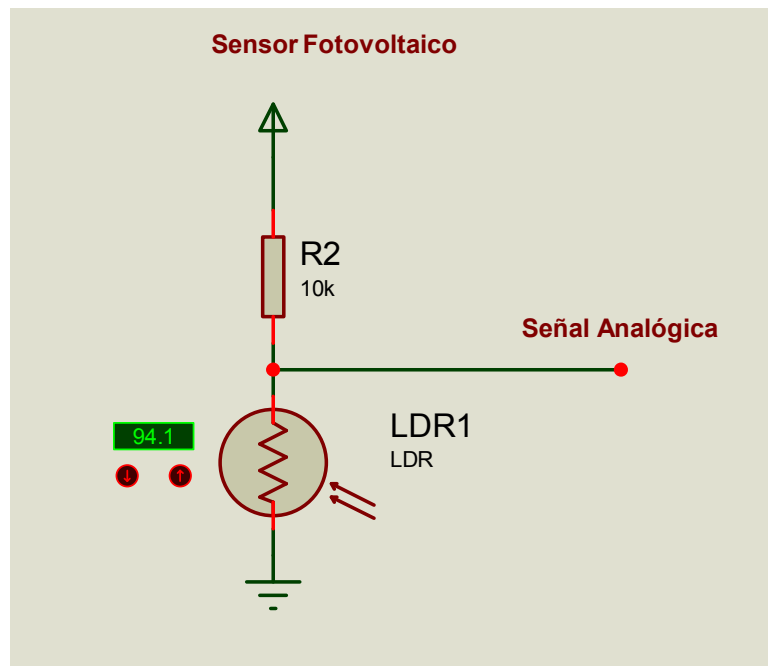


Figura 88: Diagrama de Simulación Detector de luz

En la siguiente figura, se muestra el diagrama circuital del detector de luz.

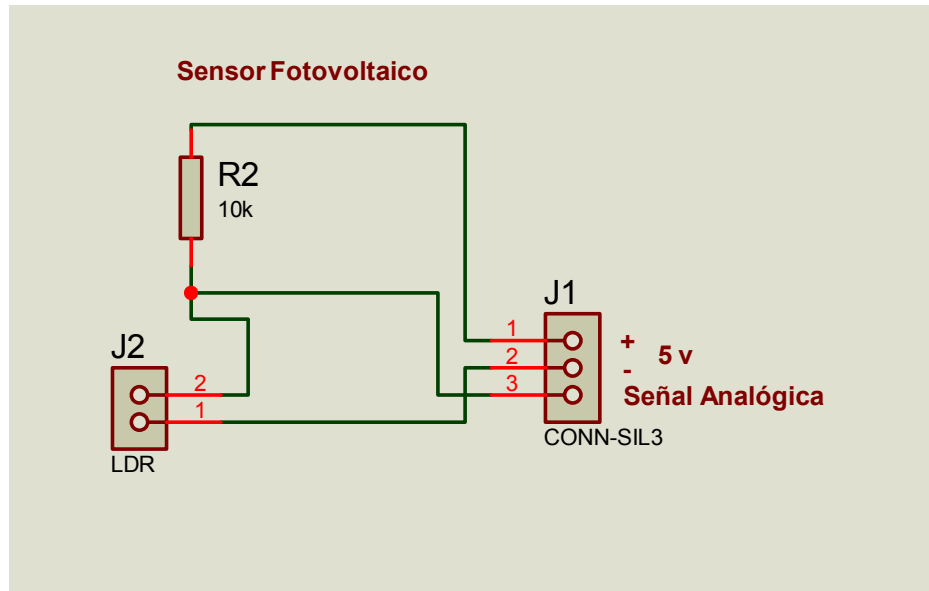


Figura 89: Diagrama Circuital Detector de luz

En la siguiente figura, se muestra el circuito diseñado en Ares.

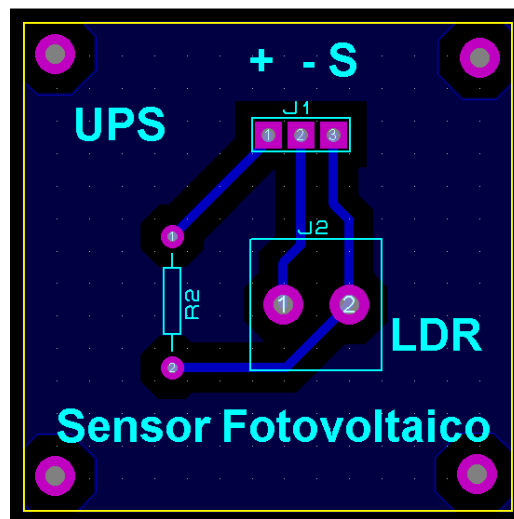


Figura 90: Circuito Detector de luz enrutado en ARES

En la figura 91, se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedara distribuido físicamente.

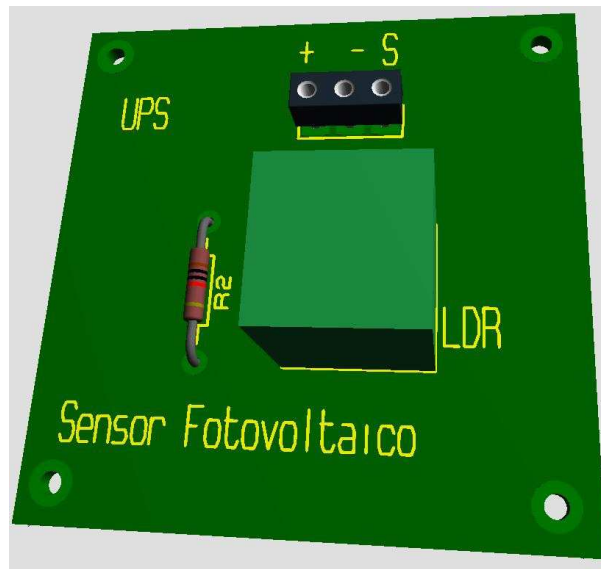


Figura 91: Vista 3D Detector de luz

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores, en este caso la figura 92, muestra la impresión de la capa cobre inferior y la figura 93, muestra la impresión de la capa seda superior que contiene los nombres de los componentes debe ser impresa en mirror.

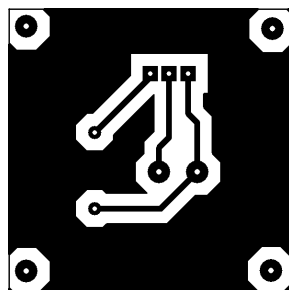


Figura 92: Bottom copper (Cobre inferior) Circuito Detector de luz

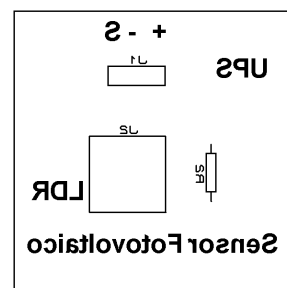


Figura 93: Top silk (Seda superior) Circuito Detector de luz

En la siguiente figura, se muestra la fotografía del circuito Detector de luz terminado.

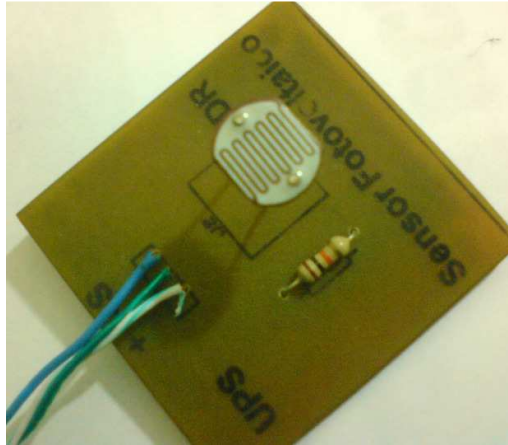


Figura 84: Fotografía del circuito Detector de luz

#### 5.2.1.5. Detector de Gas GLP

Este dispositivo envía al microcontrolador una señal analógica que varía dependiendo de la cantidad de gas GLP presente en el ambiente. Varía su resistividad incrementando o disminuyendo según la concentración de gas detectada, en la figura 95, se muestra el diagrama de simulación el detector MQ-6 envía una señal analógica que luego va a ser interpretada por el microcontrolador.

#### Materiales

- 1 Resistencia 5 k  $\Omega$
- 1 Sensor de Gas MQ-6
- 1 Capacitor de 1nF
- 1 Bornera de 3 contactos
- 1 Baquelita de cobre de 4 x 4 cm



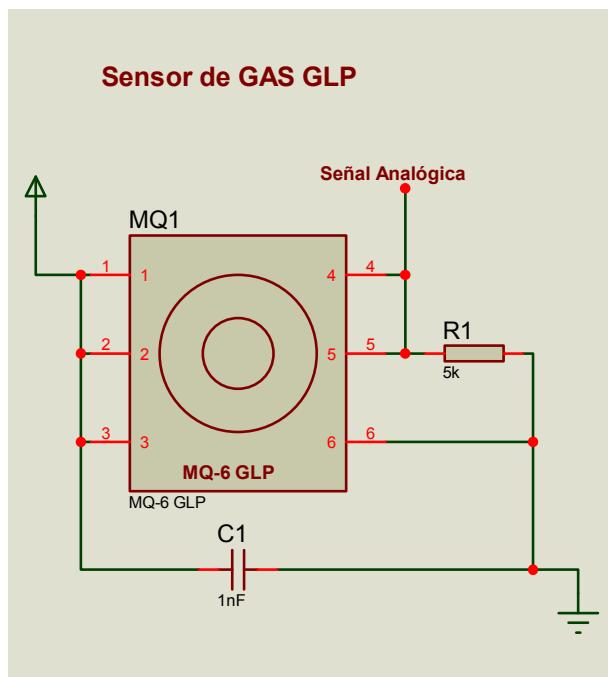


Figura 95: Diagrama de Simulación Detector de gas GLP

En la siguiente figura, se muestra el diagrama circuital del Detector de gas GLP

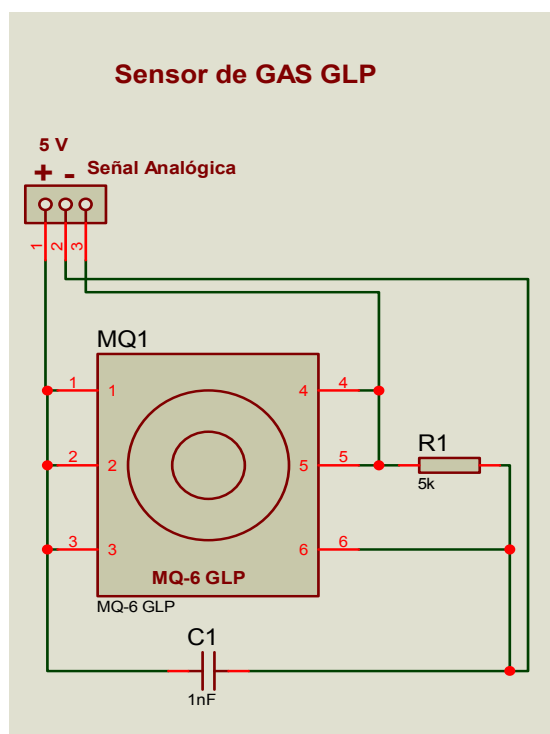


Figura 96: Diagrama Circuital del Detector de gas GLP

En la siguiente figura, se muestra el circuito diseñado en Ares.

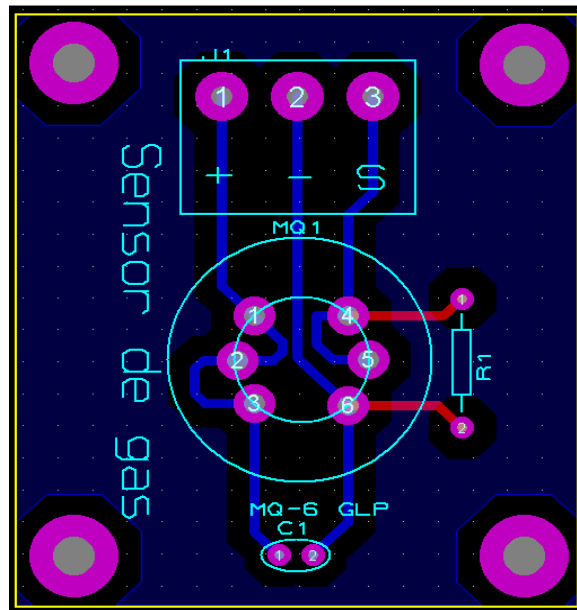


Figura 97: Circuito Detector de gas GLP enrutado en ARES

En la figura 98, se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedara distribuido físicamente.

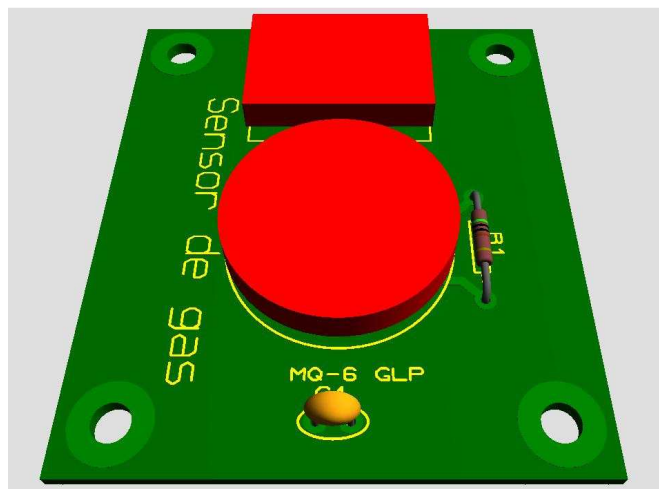
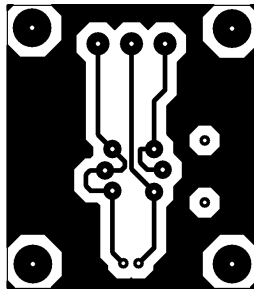
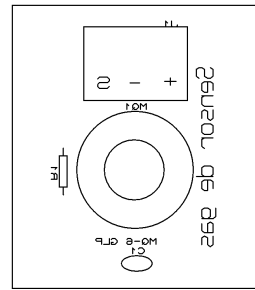


Figura 98: Vista 3D Detector de gas GLP

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 99 muestra la impresión de la capa cobre inferior y la figura 100 muestra la impresión de la capa seda superior que contiene los nombres de los componentes debe ser impresa en mirror.

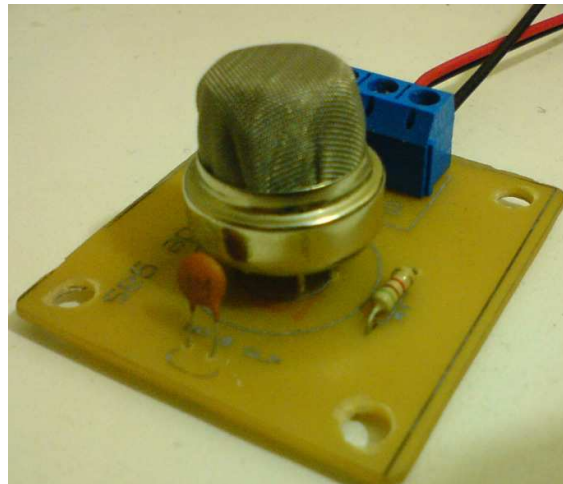


**Figura 99: Bottom copper (Cobre inferior) Circuito Detector de GAS**



**Figura 100: Top silk (Seda superior) Circuito Detector de GAS**

En la siguiente figura, se muestra la fotografía del circuito Detector de gas GLP terminado.



**Figura 101: Fotografía del circuito Detector de gas GLP**

#### 5.2.1.6. Inversor de giro Motor DC

El microcontrolador ubicado en el módulo Xbee1 activa dos relés, uno de apertura y otro de cierre de la cortina uno a la vez, la idea es hacer funcionar un motor DC en varios sentidos, es decir, invertir su giro.

Para esto es necesario invertir la polaridad, cambiar positivo y negativo, hay varias formas de hacerlo como por ejemplo usar un puente H, pero el motor consume hasta 5 Amperios lo que hace difícil encontrar un puente H que soporte esta cantidad de corriente, por lo que se plantea usar relés que soportan hasta 10 amperios.

En la figura 102, se muestra el diagrama circuital del inversor de giro de motor DC, desarrollado a base de relés en el centro hay un elemento cambio de giro, que lo único que hace es unir sus pines como se muestra en la figura.

Materiales	
4	Relés 12v - 10 A 110V
5	Bornera 2 contactos
1	Baquelita 10 x 10

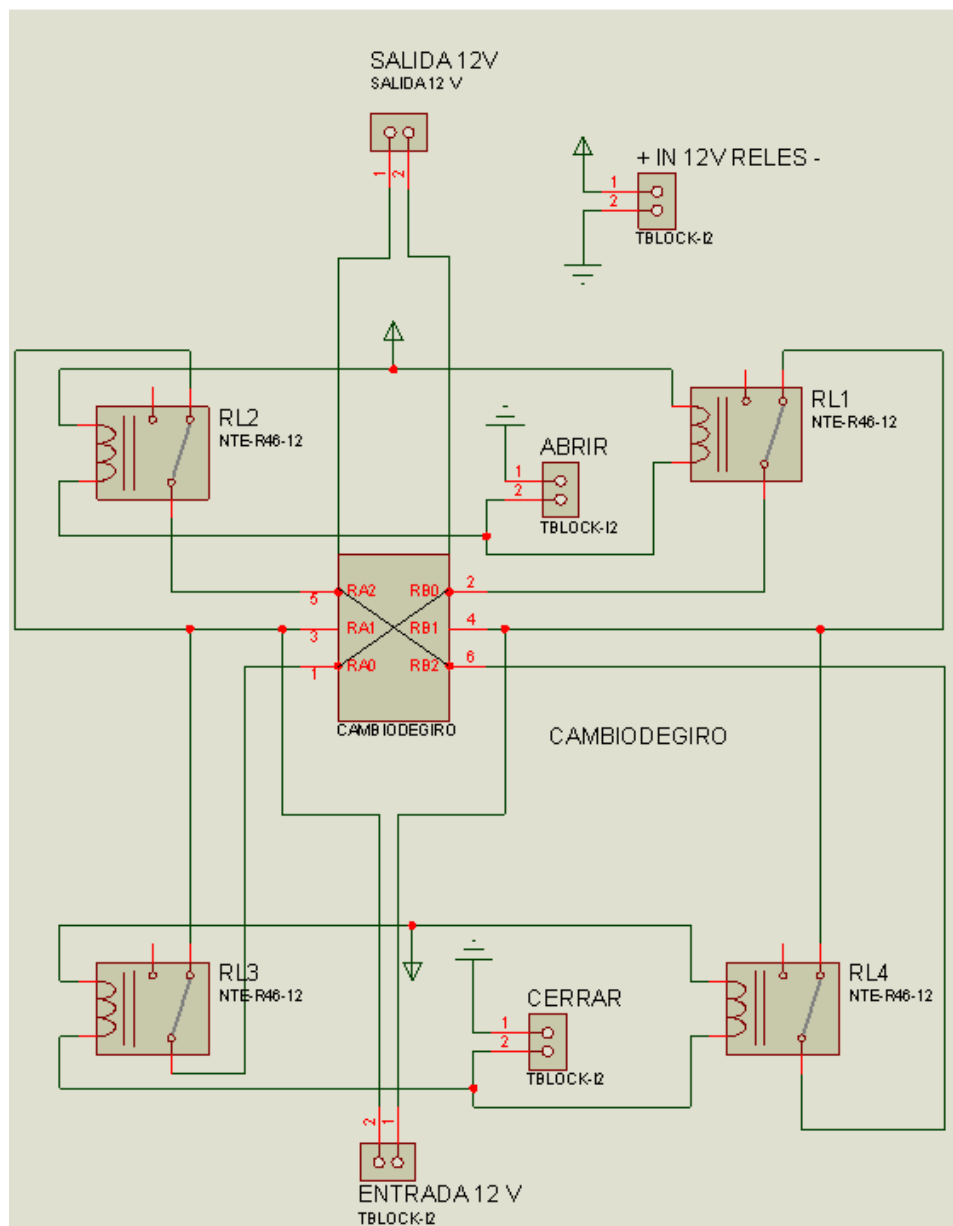


Figura 102: Diagrama circuital Inversor de giro motor DC

En la siguiente figura, se muestra el circuito diseñado en Ares.

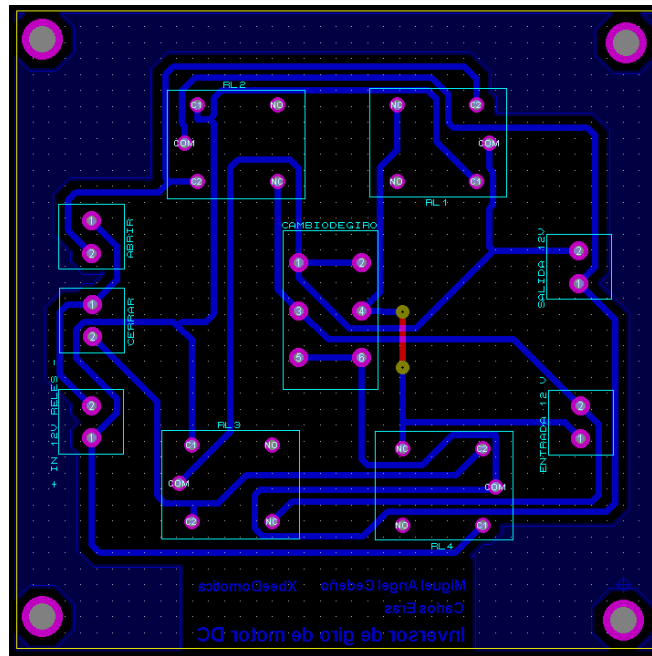


Figura 103: Circuito inversor de giro de motor DC enrutado en ARES

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 104, muestra la impresión de la capa cobre inferior y la figura 105, muestra la impresión de la capa seda superior que contiene los nombres de los componentes, debe ser impresa en mirror.

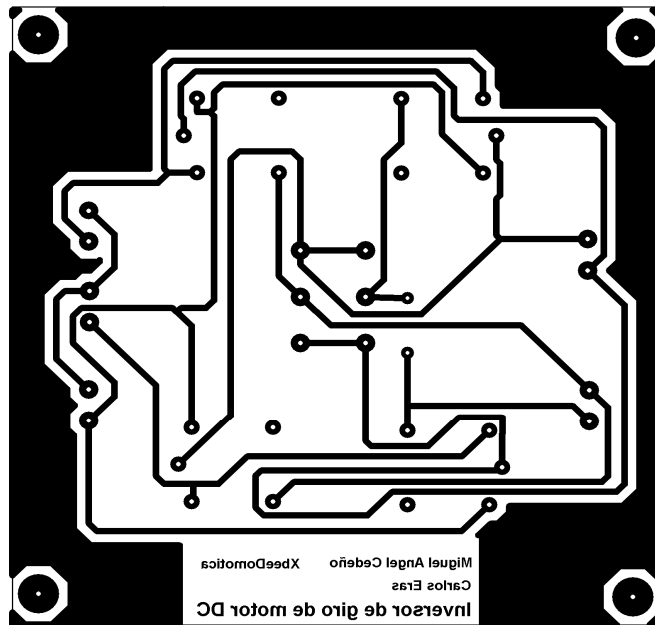


Figura 104: Bottom copper (Cobre inferior) Circuito inversor de giro de motor DC

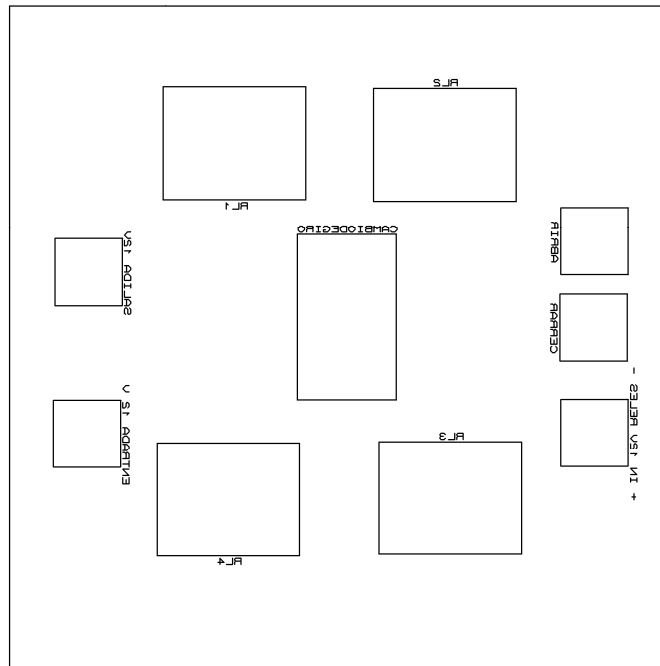


Figura 105: Top silk (Seda superior) Circuito inversor de giro de motor DC

En la siguiente figura, se muestra la fotografía del circuito inversor de giro de motor DC terminado.



Figura 106: Fotografía circuito Inversor de giro motor DC

### 5.2.2. MÓDULO XBEE1

Es el más complejo de todos los módulos que se van a construir, maneja entradas analógicas y digitales, controla la apertura y cierre de cortina con sus respectivos sensores magnéticos de fines de ciclo.

Para el PCB se usa una placa de doble cara debido a la cantidad de componentes y al reducido espacio, es muy complejo desarrollarlo en una sola capa.

#### Materiales

5	Relés 12v - 10 A 110V
9	Bornera 2 contactos
1	Bornera 3 contactos
1	Regulador 3.3v LD33CV
1	Regulador 5v LM7805
14	Resistencias 4.7 K $\Omega$
4	Resistencias 10 K $\Omega$
9	Resistencias 330
9	Diodos Rectificador 1N4007 1A/1000V
5	Transistores 2N3904
2	Condensadores Cerámico 22pf/50v
1	Oscilador de Cristal 4MHz
1	Disipador Regulador 5v
1	Jack de Fuente
1	Broche Batería 9v
1	Batería 9v
2	Socket Xbee 10 pines
1	Pulsador 2 pines
14	Conectores hembra 2 pines
9	Leds



4	Pulsadores Grandes
1	Pic 18F452
1	Xbee serie 2
1	Zócalo 40 p
1	Detector de Movimiento
1	Detector de Humo
1	Bocina 12v CBZ-50
2	Sensores Magnéticos
1	Sensor Fotovoltaico
1	Caja Metálica
4	Postes 15 mm
4	Tornillos Postes
5	Cables R/N
1	Fuente 12v
1	Fibra doble lado 10 x 15

En la siguiente figura, se muestra el diagrama de simulación del Módulo Xbee1, por motivos de simulación se usa un Hyperterminal con dos max-232 en reemplazo del módulo Xbee series 2, porque en el simulador ISIS no existe este componente inalámbrico.

En la siguiente figura se muestra el diagrama circuital del Módulo Xbee1.

Xbee 1 Mov1, DetecHumo, Sala, Bocina, Sensor fotovoltaico, Cortina

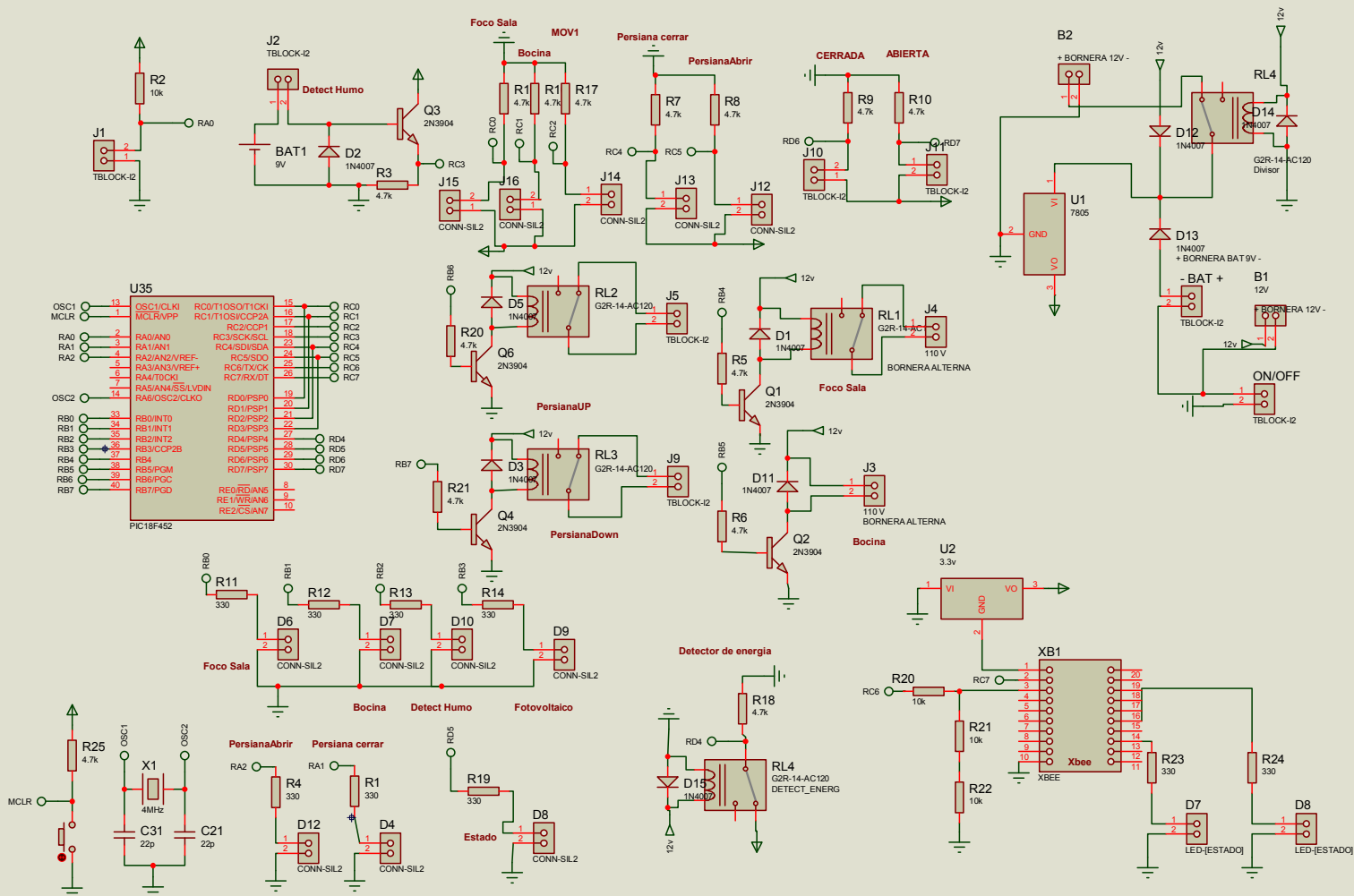


Figura 108: Diagrama Circuital Módulo Xbee1

En Proteus no existe el componente Xbee, se debe lo crear con la ayuda del datasheet, usando sus medidas y número de pines.

En la siguiente figura, se muestra el módulo Xbee1 enrutado en ARES.

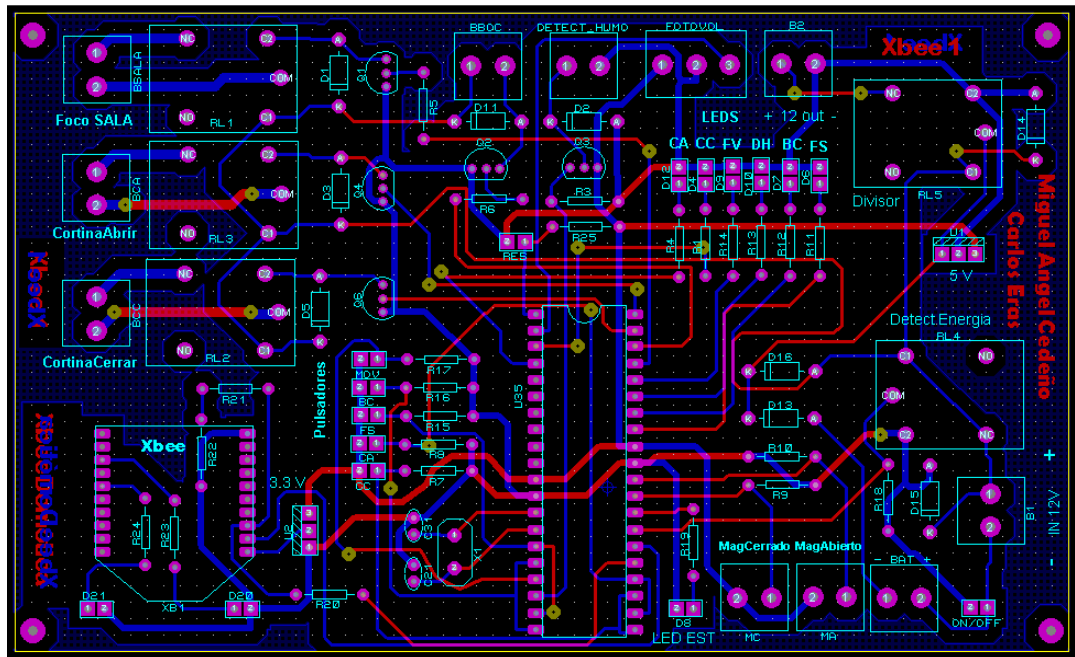
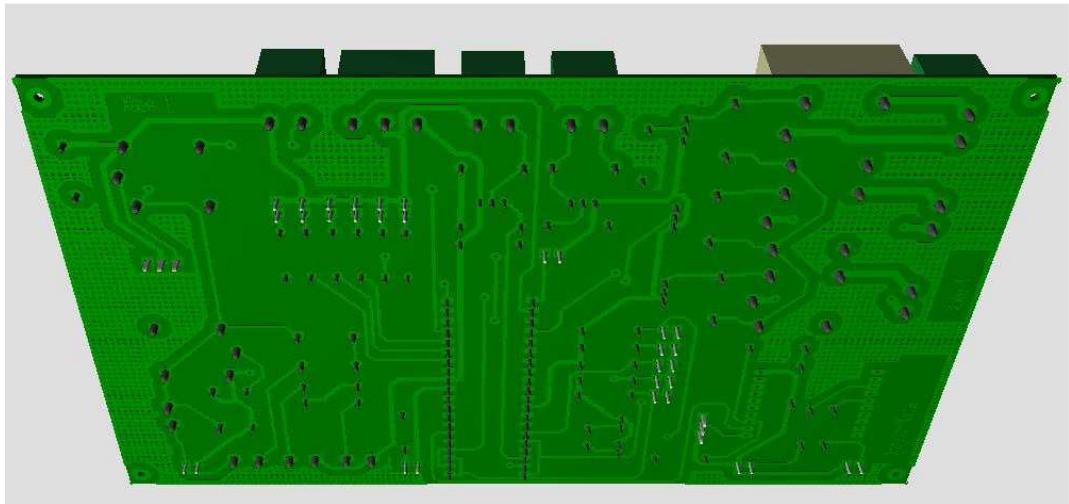
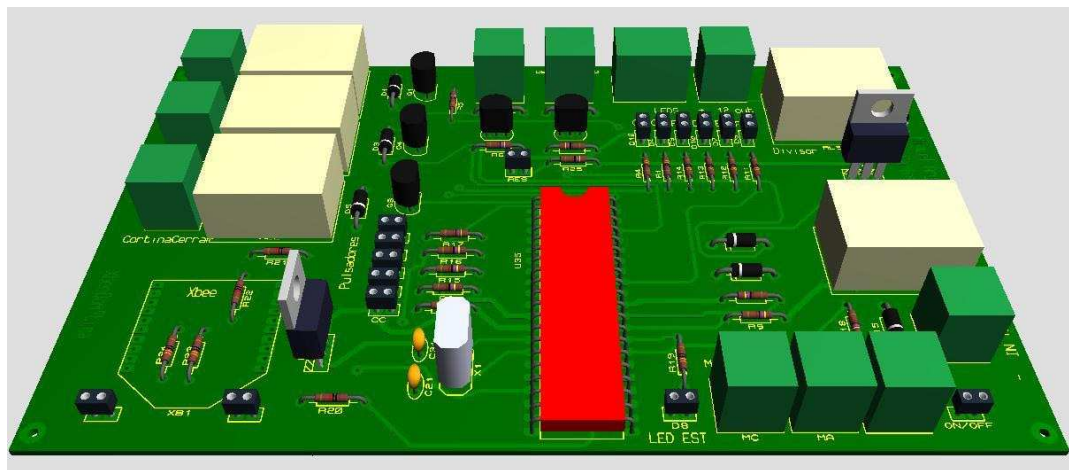


Figura 109: Circuito Módulo Xbee1 enrutado en ARES

En la figura 110 y la figura 111 se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedara distribuido físicamente.



**Figura 110: Vista 3D inferior Módulo Xbee1**



**Figura 111: Vista 3D superior Módulo Xbee 1**

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 112 muestra la impresión de la capa cobre inferior, la figura 113 muestra la impresión de la capa cobre superior y la figura 114 muestra la impresión de la capa seda superior que contiene los nombres de los componentes, las capas superiores deben ser impresas en mirror.

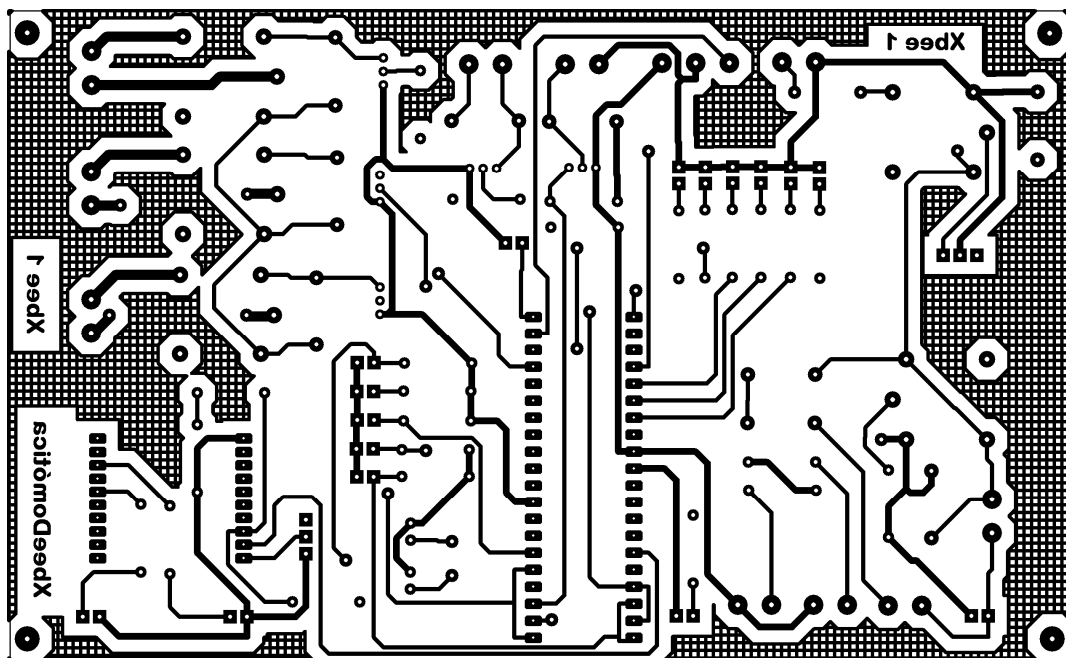


Figura 112: Bottom copper (Cobre inferior) Circuito Módulo Xbee1

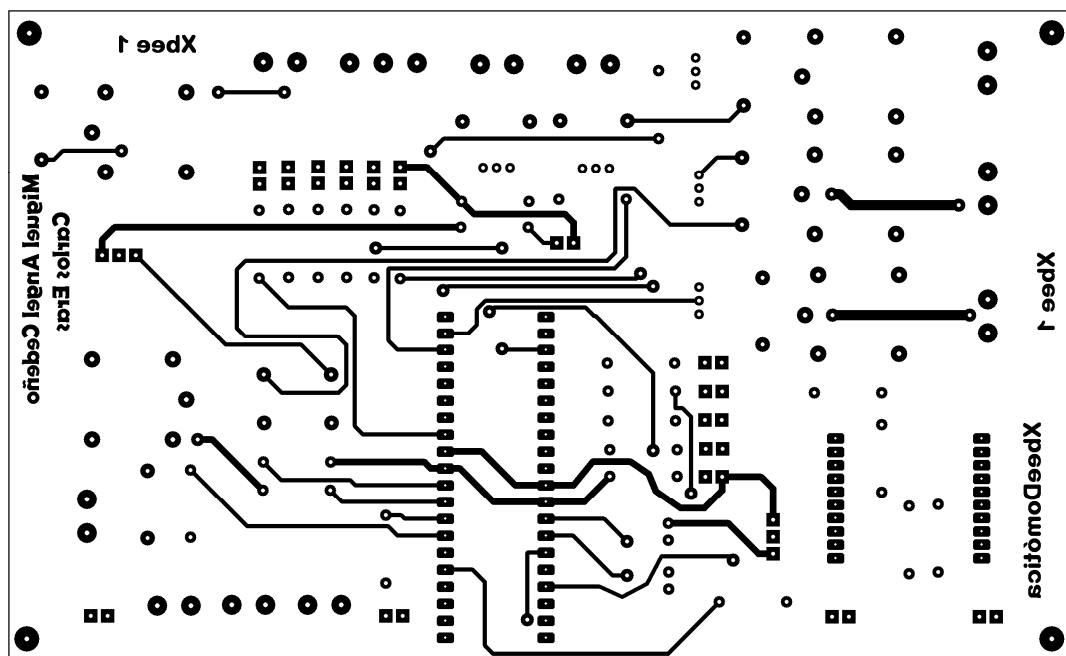
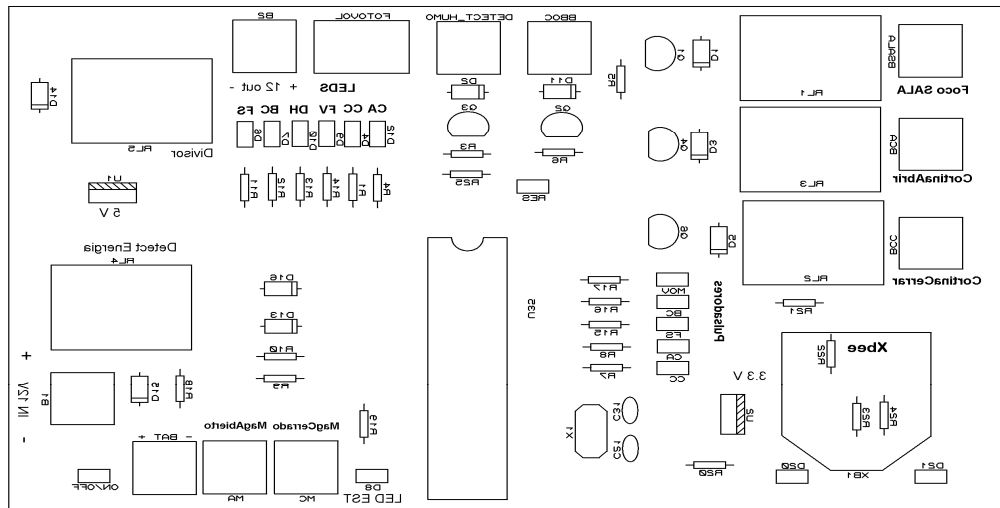
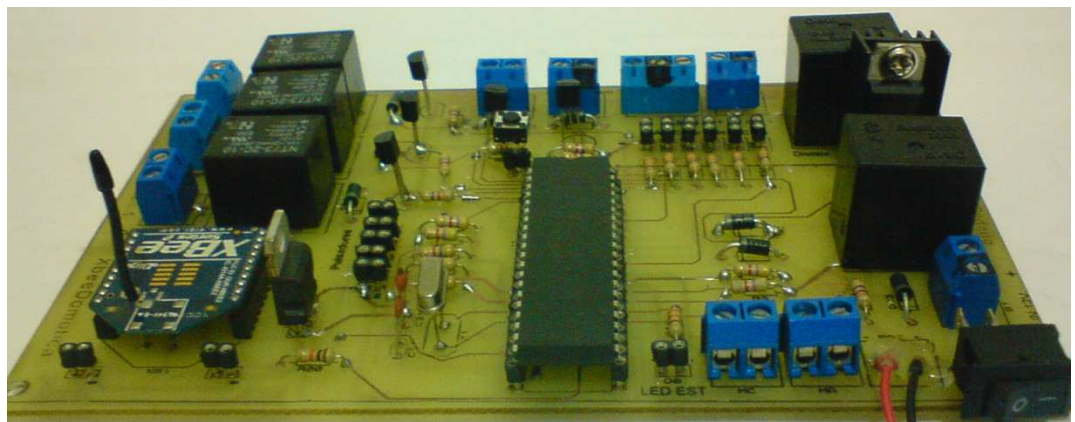
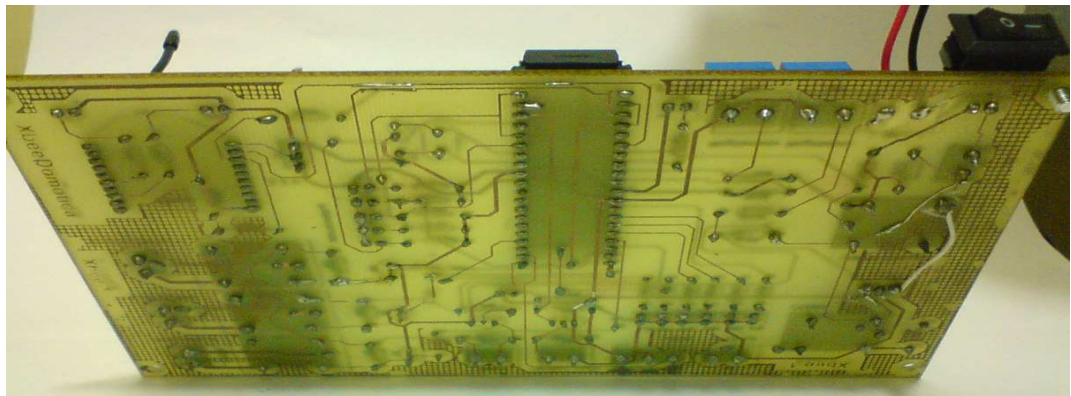


Figura 113: Top copper (Cobre superior) Circuito Módulo Xbee 1



En las siguientes figuras 115 y 116, se muestra la fotografía del circuito Módulo 1 terminado.



### 5.2.3. MÓDULO XBEE2

Para el PCB se usa una placa de doble cara debido a la cantidad de componentes y al reducido espacio, es muy complejo desarrollarlo en una sola capa.

#### Materiales

4	Relés 12v - 10 A 110V
4	Borneras 2 contactos
2	Borneras 3 contactos
1	Regulador 3.3v LD33CV
1	Regulador 5v LM7805
7	Resistencias 4.7 K $\Omega$
4	Resistencias 10 K $\Omega$
8	Resistencias 330 $\Omega$
6	Diodos Rectificador 1N4007 1A/1000V
2	Transistores 2N3904
2	Condensadores Cerámico 22pf/50v
1	Oscilador de Cristal 4MHz
1	Disipador Regulador 5v
1	Jack de Fuente
1	Broche Batería 9v
1	Batería 9v
2	Socket Xbee 10 pines
1	Pulsador 2 pines
8	Conectores hembra 2 pines
7	Leds
2	Pulsadores Grandes
1	Pic 18F452
1	Xbee serie 2
1	Zócalo para PIC de 40 pines



1	Detector de Movimiento
1	Sensor de Gas GLP
1	Buzzer Activo 5v
1	Caja Metálica
4	Postes 15mm
4	Tornillos Postes
1	Fuente 12 v
1	Fibra doble lado 10 x 15

En la siguiente figura, se muestra el diagrama de simulación del Módulo Xbee2, por motivos de simulación se usa un Hyperterminal con dos max-232 en reemplazo del módulo Xbee series 2, porque en el simulador ISIS no existe este componente inalámbrico.

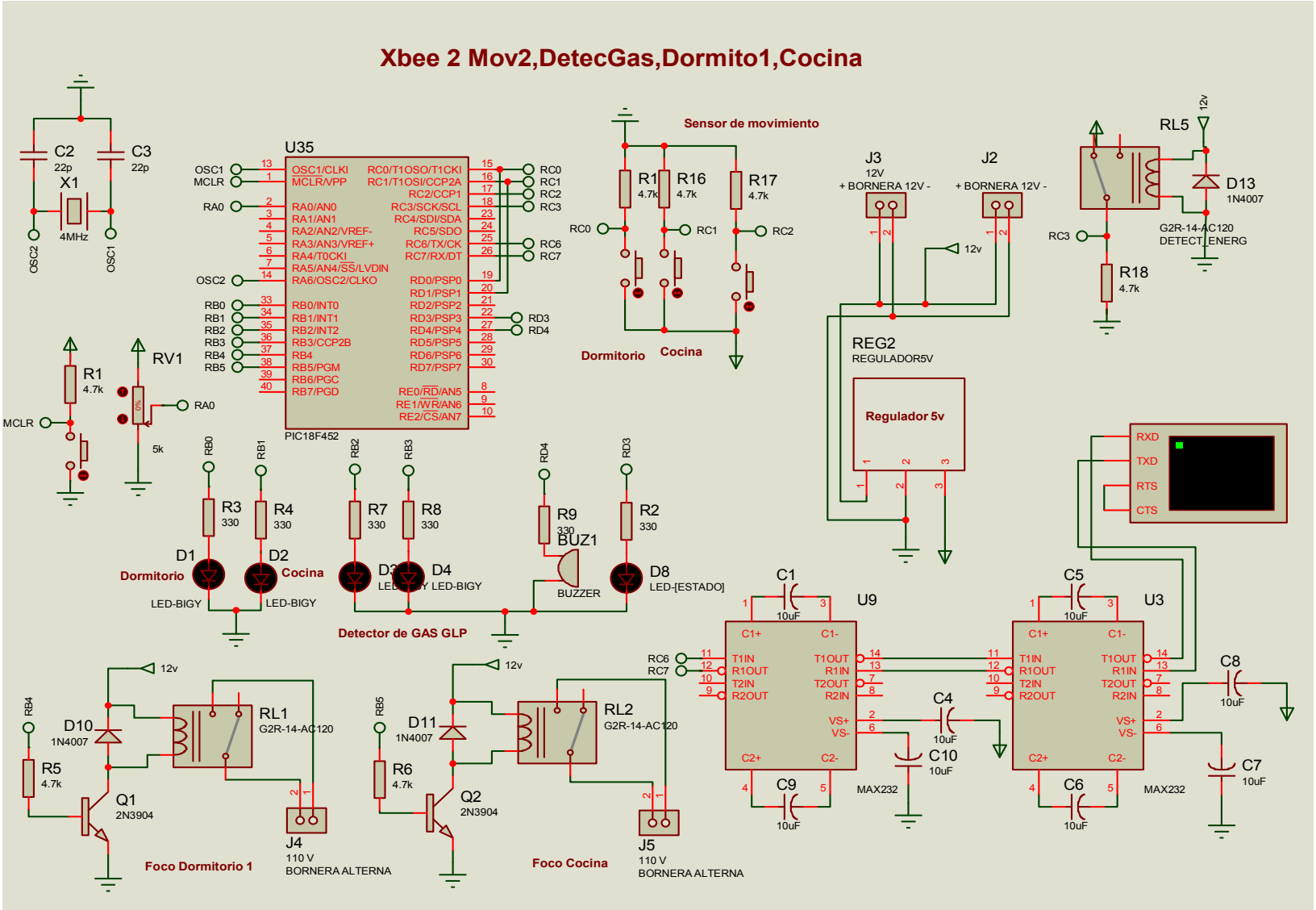


Figura 117: Diagrama de Simulación Módulo Xbee 2

En la siguiente figura, se muestra el diagrama circuital del Módulo Xbee2

## Xbee 2 Mov2, DetecGas, Dormito1, Cocina

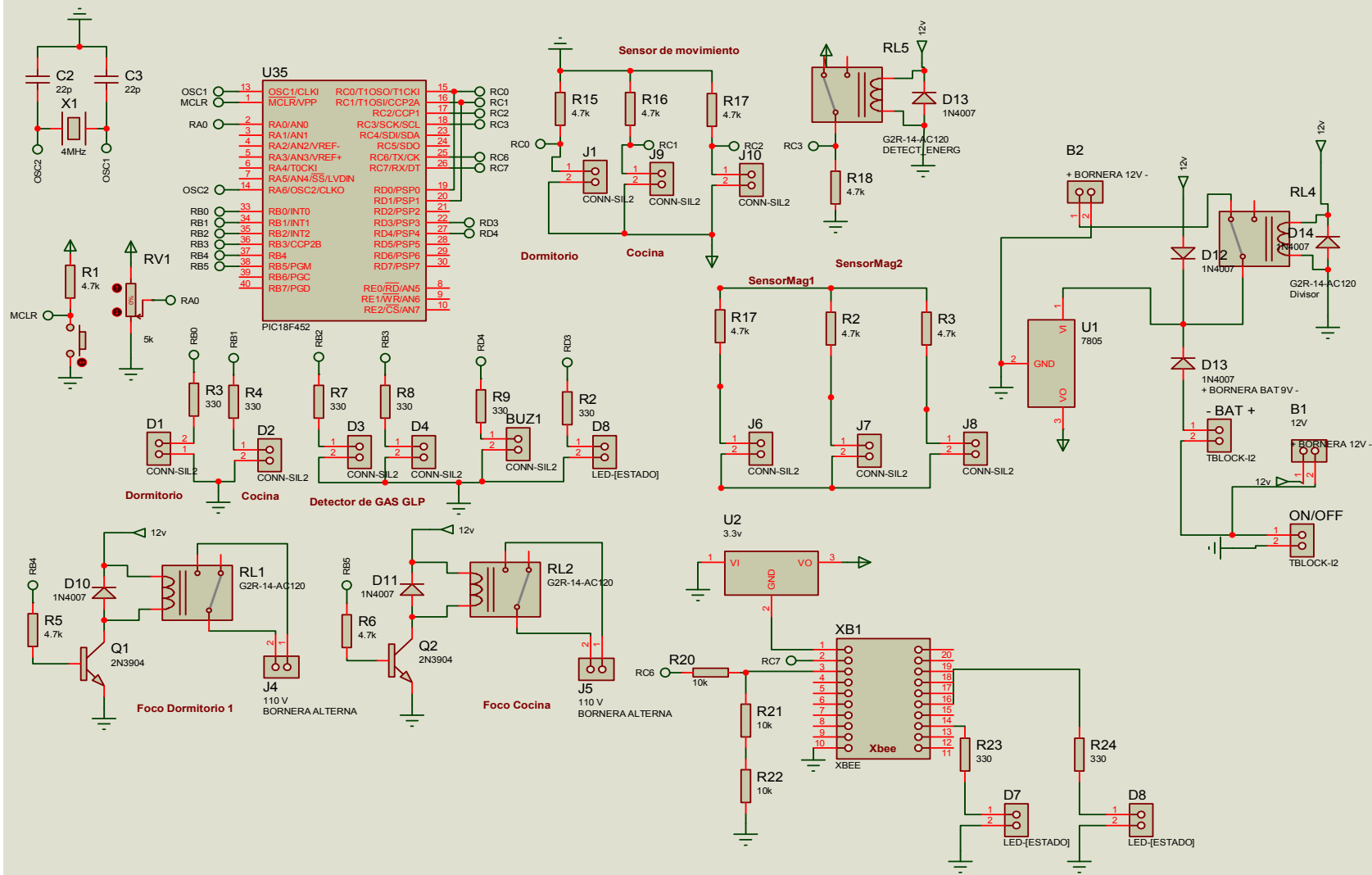


Figura 118: Diagrama Circital Módulo Xbee 2

En Proteus no existe el componente Xbee, se debe lo crear con la ayuda del datasheet, usando sus medidas y número de pines.

En la siguiente figura, se muestra el módulo Xbee2 enrutado en ARES.

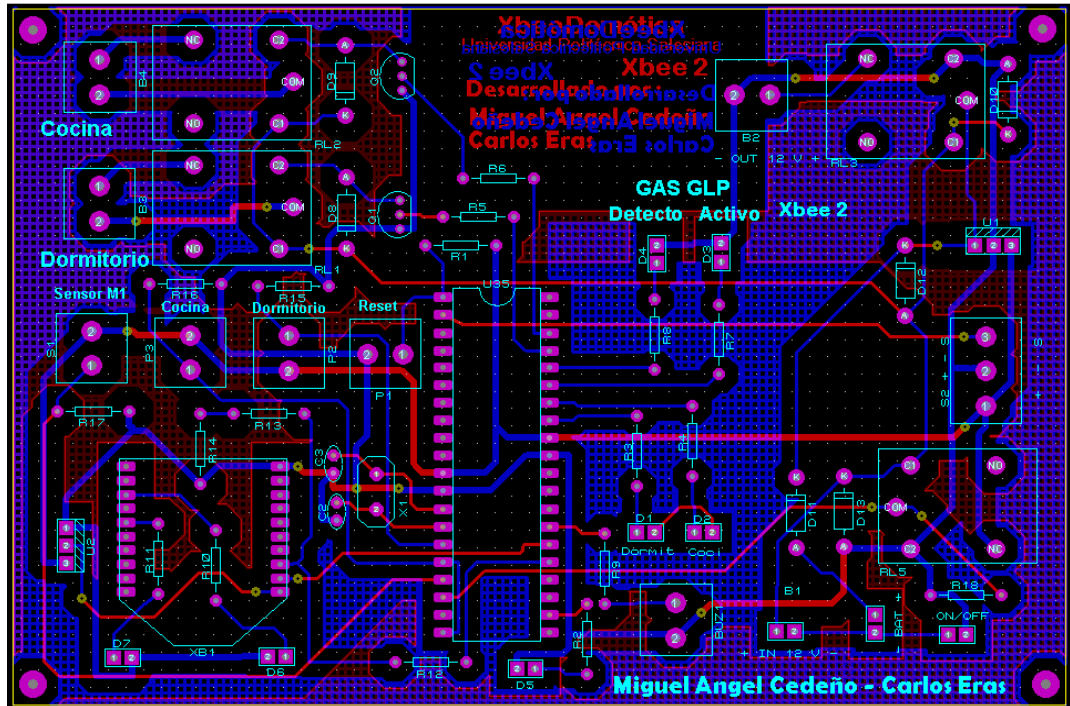


Figura 119: Circuito Módulo Xbee 2 enrutado en ARES

En la figura 120 y la figura 121 se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedará distribuido físicamente.

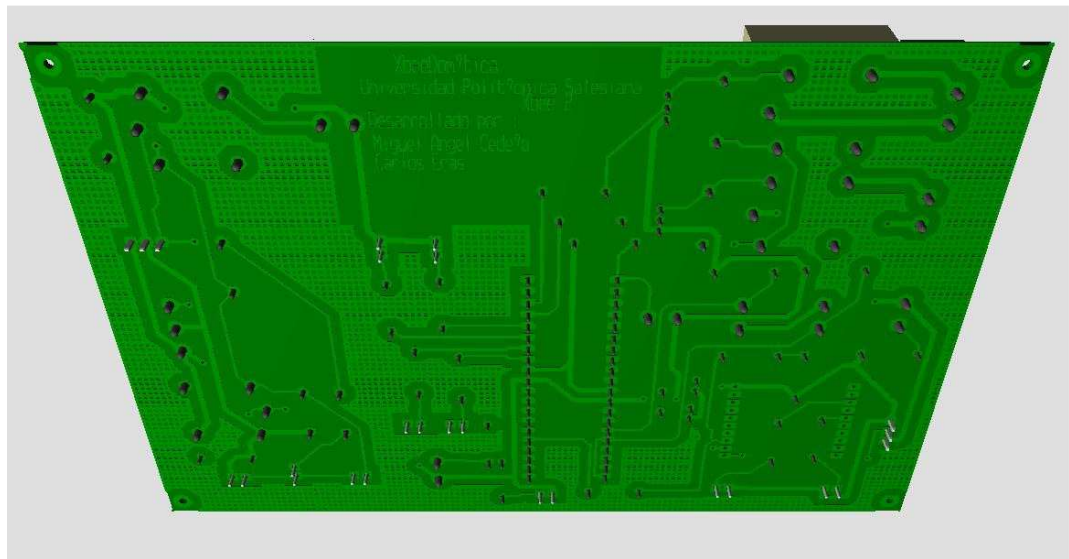


Figura 120: Vista 3D inferior Módulo Xbee 2

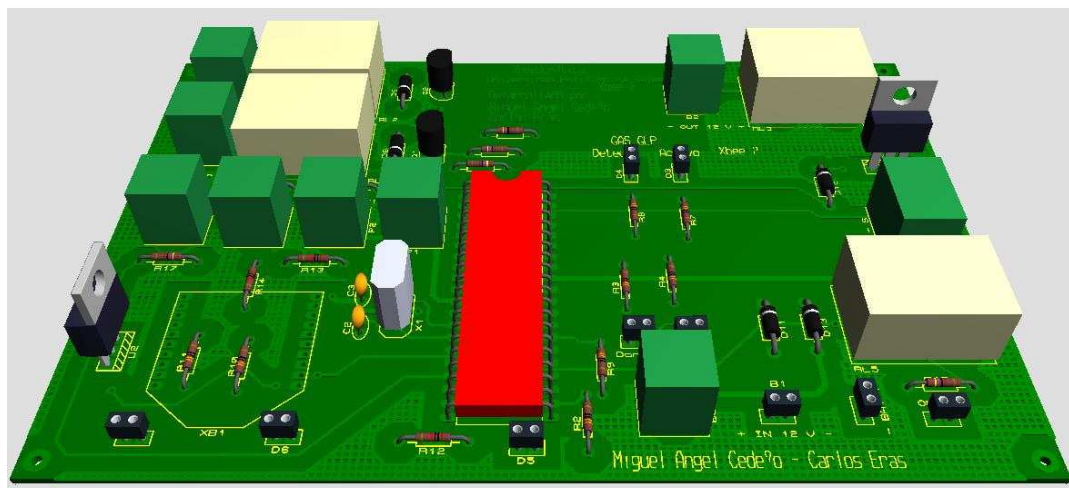


Figura 121: Vista 3D superior Módulo Xbee 2

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 122 muestra la impresión de la capa cobre inferior, la figura 123 muestra la impresión de la capa cobre superior y la figura 124 muestra la impresión de la capa seda superior que contiene los nombres de los componentes, las capas superiores deben ser impresas en mirror.

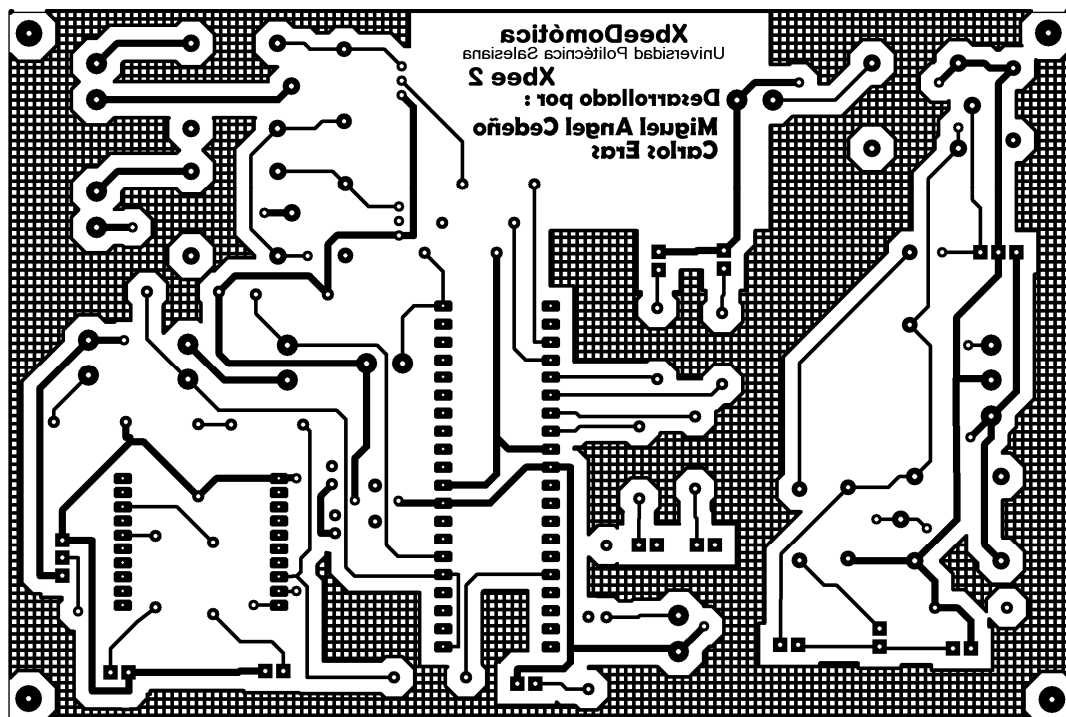


Figura 122: Bottom copper (Cobre inferior) Circuito Módulo Xbee 2

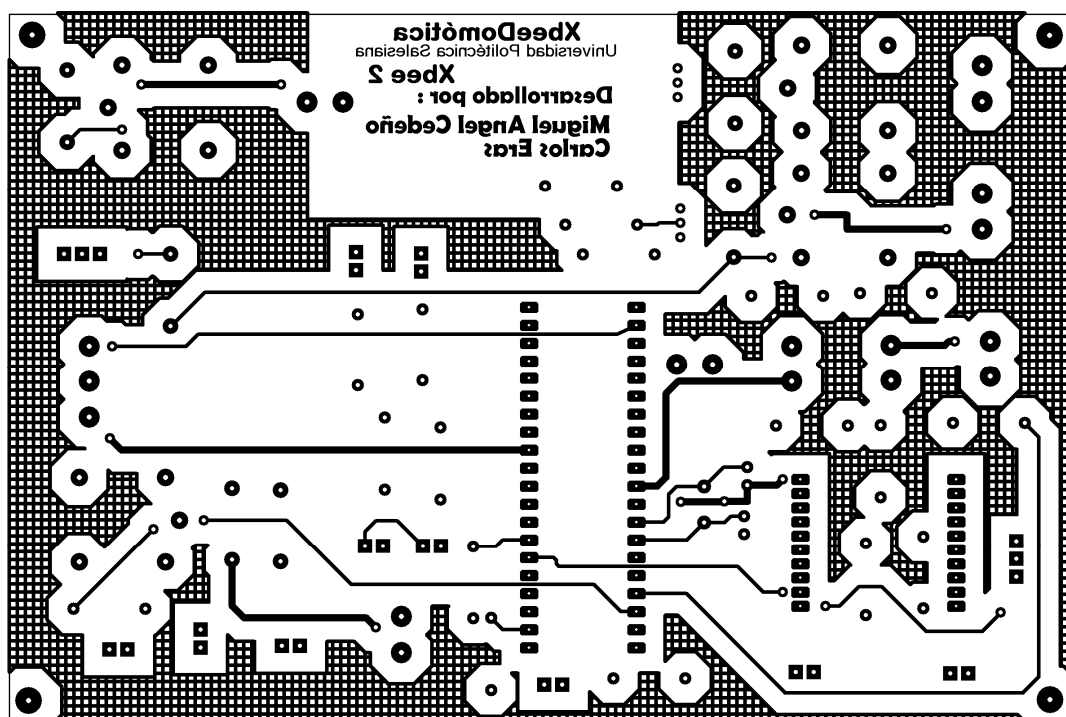
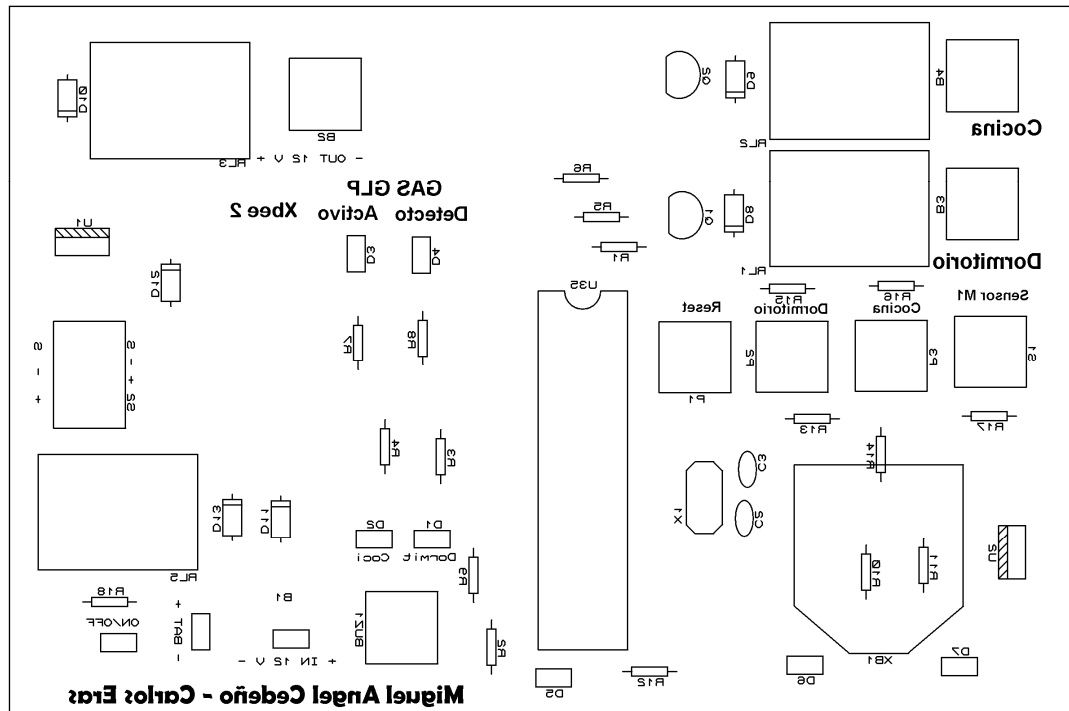


Figura 123: Top copper (Cobre superior) Circuito Módulo Xbee 2



En las siguientes figuras, se muestran las fotografías del circuito Módulo 2 terminado.

#### 5.2.4. MÓDULO XBEE3

Para el PCB se usa una placa de doble cara debido a la cantidad de componentes y al reducido espacio, es muy complejo desarrollarlo en una sola capa.

##### Materiales

5	Relés 12v - 10 A 110V
7	Bornas 2 contactos
1	Regulador 3.3v LD33CV
1	Regulador 5v LM7805
10	Resistencias 4.7K $\Omega$
3	Resistencias 10K $\Omega$
8	Resistencias 330 $\Omega$
7	Diodos Rectificador 1N4007 1A
3	Transistores 2N3904
2	Condensadores Cerámico 22pf/50v
1	Oscilador de Cristal 4MHz
1	Disipador Regulador 5v
1	Jack de Fuente
1	Broche Batería 9v
1	Batería 9v
2	Socket Xbee 10 pines
1	Pulsador 2 pines
11	Conectores hembra 2 pines
8	Leds
3	Pulsadores Grandes
1	Pic 18F452
1	Xbee serie 2
1	Zócalo Pic 40 pines



2	Sensores Magnéticos
1	Caja Metálica
4	Postes 15mm
4	Tornillos Postes
1	Fuente 12 v 800mA
1	Fibra doble lado 10 x 15

En la siguiente figura, se muestra el diagrama de simulación del Módulo Xbee 3, por motivos de simulación se usa un Hyperterminal con dos max-232 en reemplazo del módulo Xbee series 2, porque en el simulador ISIS no existe este componente inalámbrico.

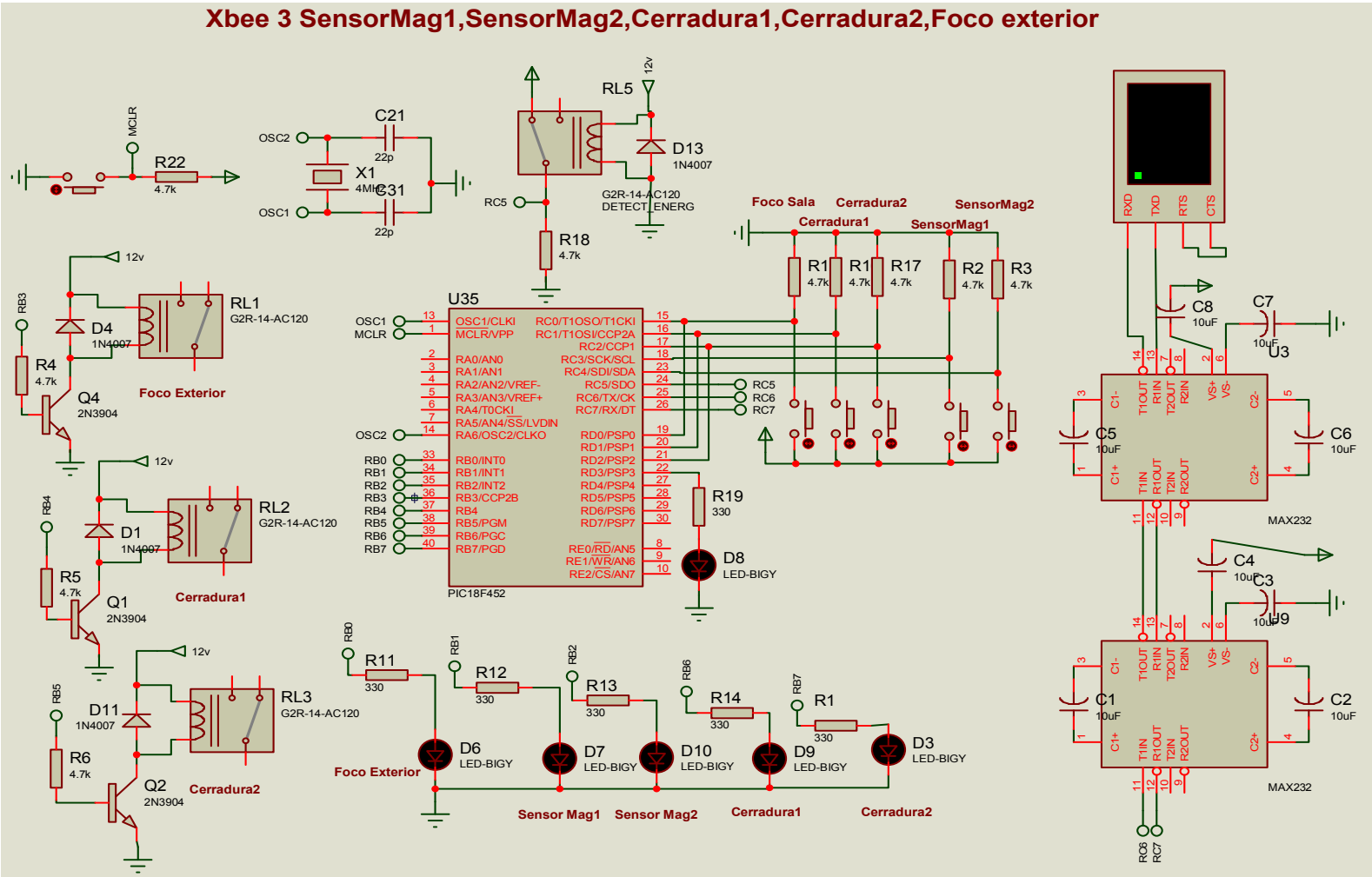


Figura 126: Diagrama de Simulación Módulo Xbee 3

En la siguiente figura, se muestra el diagrama circuital del Módulo Xbee3



En Proteus no existe el componente Xbee, se debe lo crear con la ayuda del datasheet, usando sus medidas y número de pines.

En la siguiente figura, se muestra el módulo Xbee 3 enrutado en ARES.

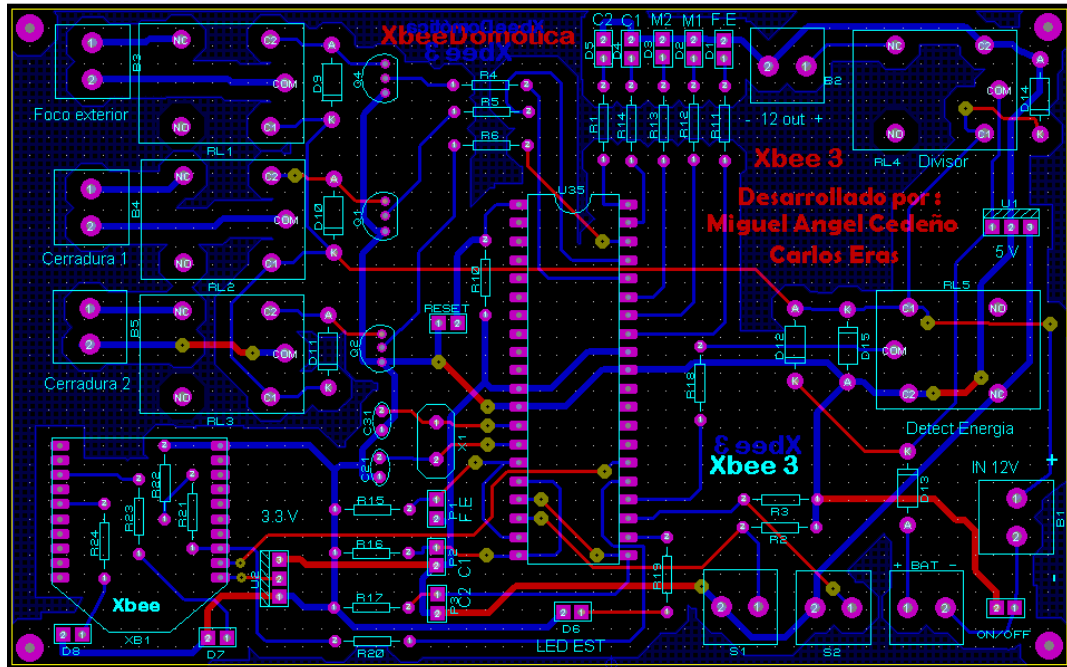


Figura 128: Circuito Módulo Xbee 3 enrutado en ARES

En las siguientes figuras, se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedara distribuido físicamente.

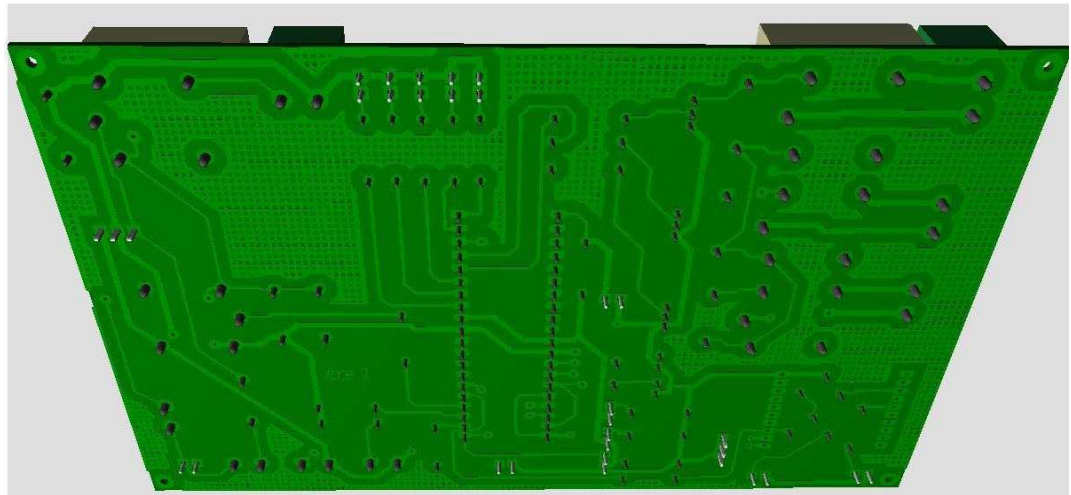


Figura 129: Vista 3D inferior Módulo Xbee 3

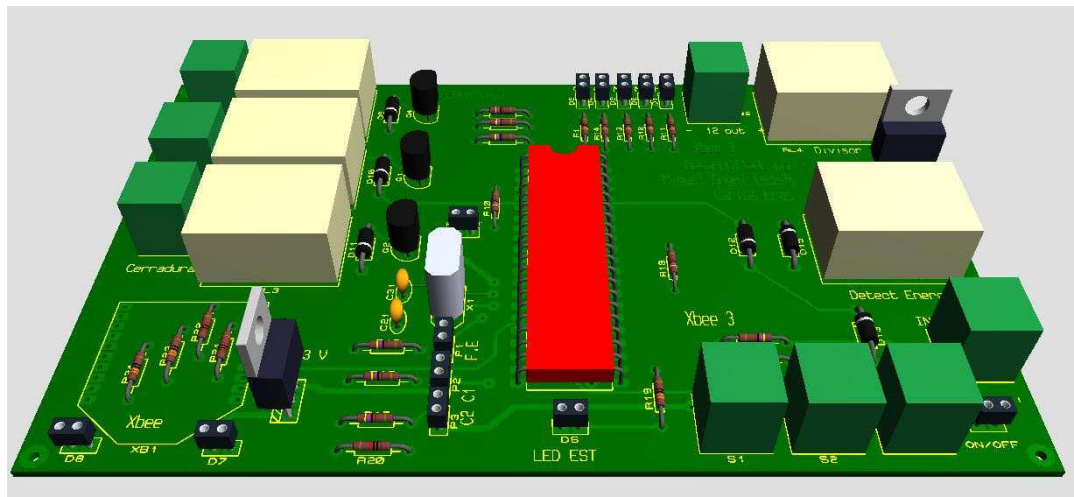


Figura 130: Vista 3D superior Módulo Xbee 3

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 131 muestra la impresión de la capa cobre inferior, la figura 132 muestra la impresión de la capa cobre superior y la figura 133 muestra la impresión de la capa seda superior que contiene los nombres de los componentes, las capas superiores deben ser impresas en mirror.

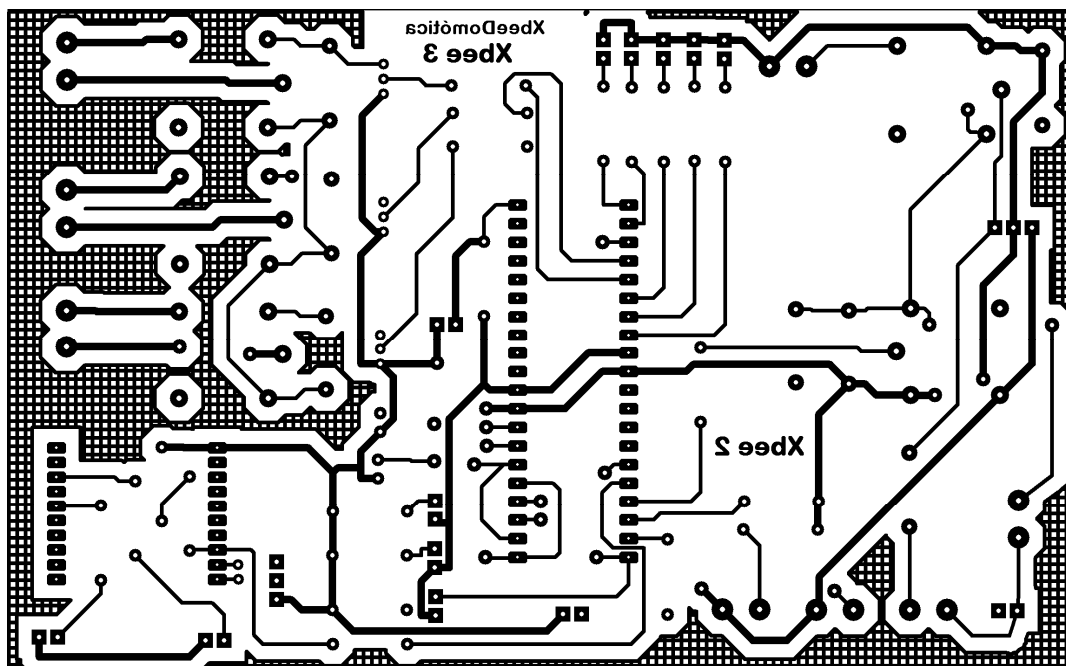


Figura 131: Bottom copper (Cobre inferior) Circuito Módulo Xbee 3

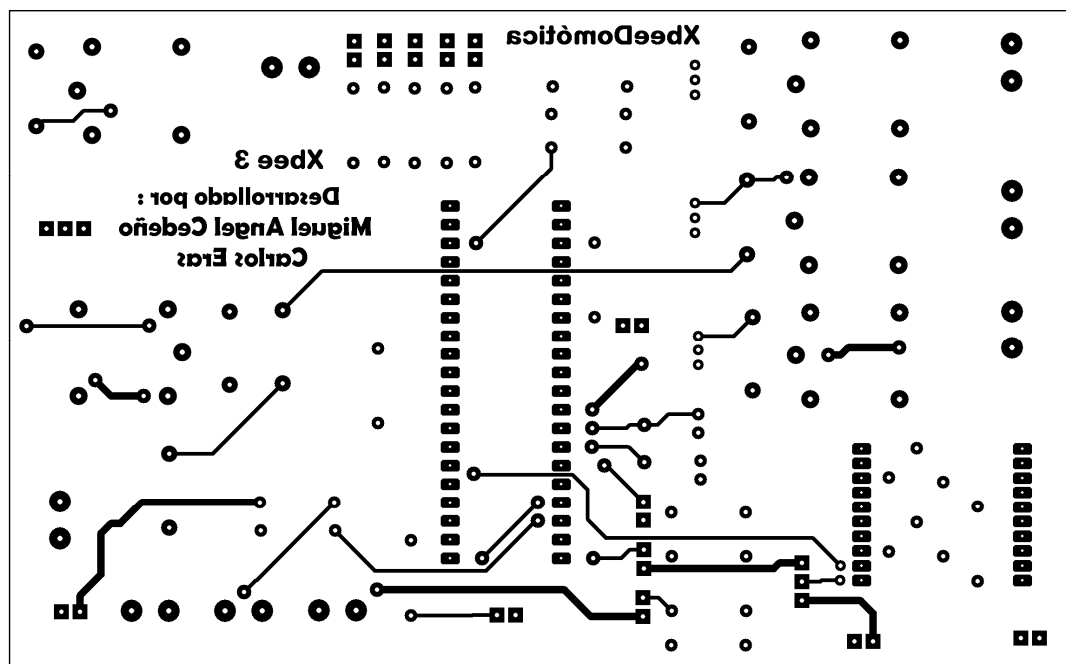
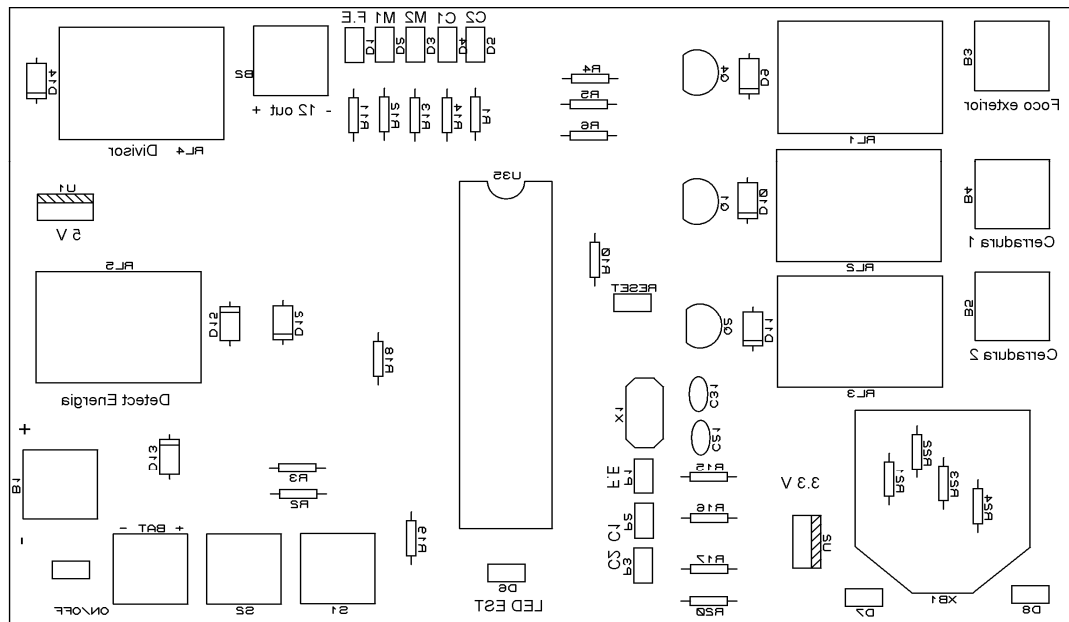
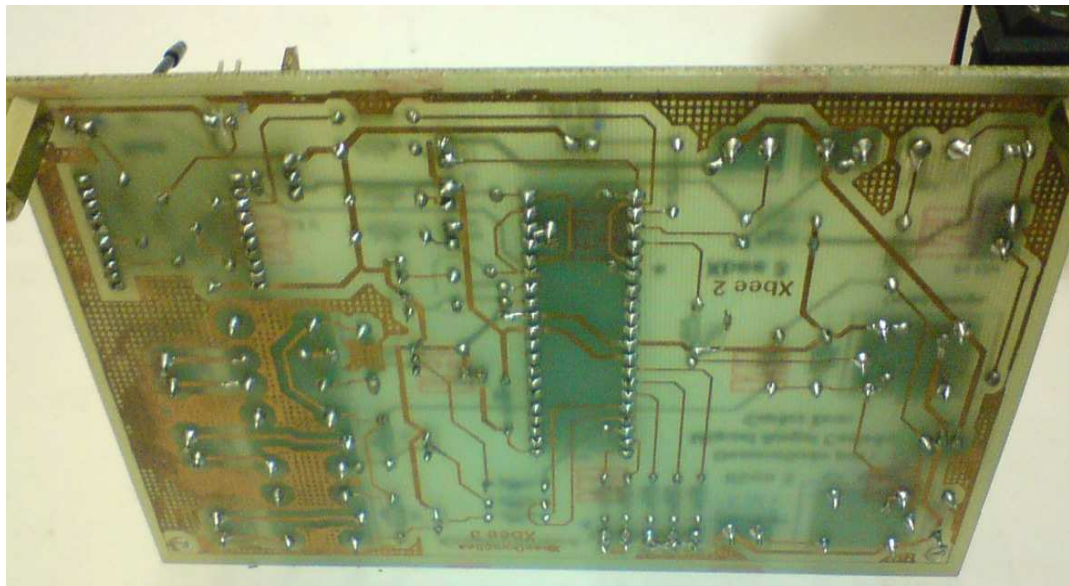


Figura 132: Top copper (Cobre superior) Circuito Módulo Xbee 3



En las siguientes figuras, se muestran las fotografías del circuito Módulo Xbee 3 terminado.



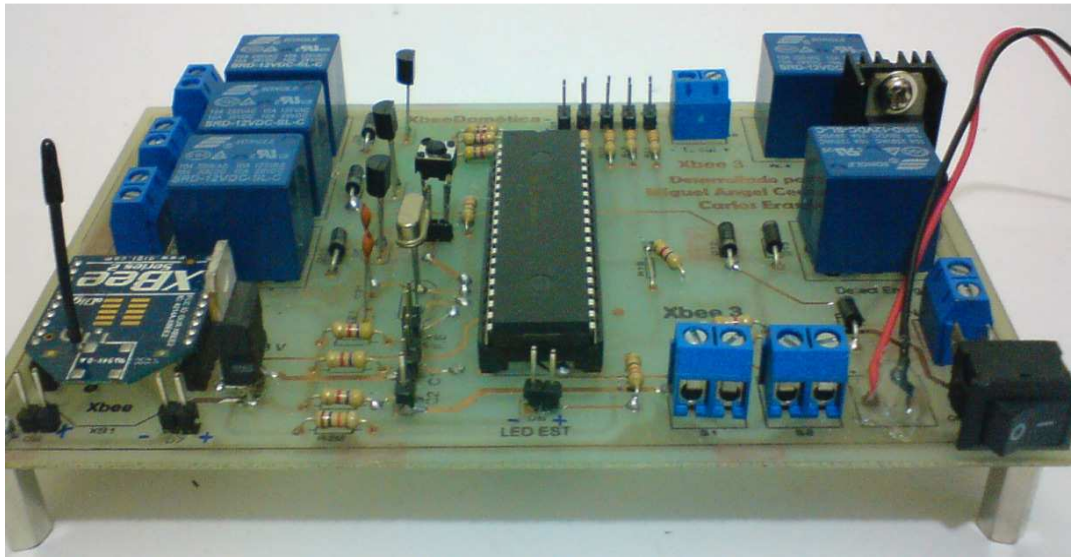


Figura 135: Fotografía superior del circuito Módulo Xbee 3

### 5.2.5. MÓDULO GLCD

Para el PCB del módulo GLCD se usa una placa de doble cara debido a la cantidad de componentes y al reducido espacio, es muy complejo desarrollarlo en una sola capa.

El panel resistivo debe ir pegado al GLCD con silicona para garantizar que no se mueva.

### Materiales

2	Borneras 2 pines
1	Regulador 3.3v LD33CV
1	Regulador 5v LM7805
4	Resistencias de 100 $\Omega$
1	Resistencias 4.7K $\Omega$
3	Resistencias 10K $\Omega$
3	Resistencias 330 $\Omega$
1	Diodos Rectificador 1N4007 1A



2	Condensadores Cerámico 22pf/50v
1	Oscilador de Cristal 4MHz
1	Disipador Regulador 5v
1	Jack de Fuente
1	Potenciómetro 10kΩ
1	Batería Recargable 6v 1,3 A
2	Socket Xbee 10 pines
8	Conectores hembra 2 pines
3	Leds
1	Pulsadores Grandes
1	Pic 18F452
1	Xbee serie 2
1	Zócalo Pic 40 pines
1	GLCD Touchscreen 240 x 128
2	Regleta 40 pines hembra
1	Bus de Datos 20 líneas
1	Bus de Datos 4 líneas
4	Postes 15mm
4	Tornillos Postes
1	Fuente 12 v 2 A
1	Fibra doble lado 10 x 15

En la siguiente figura, se muestra el diagrama de simulación del Módulo GLCD, el pin-out del GLCD 240 x 128 LM4229 de Proteus puede variar con el T6963C que se usa en este módulo, tampoco se puede simular el panel resistivo táctil por lo que la simulación solo sirve para mostrar la posición de los caracteres e imágenes del sistema.

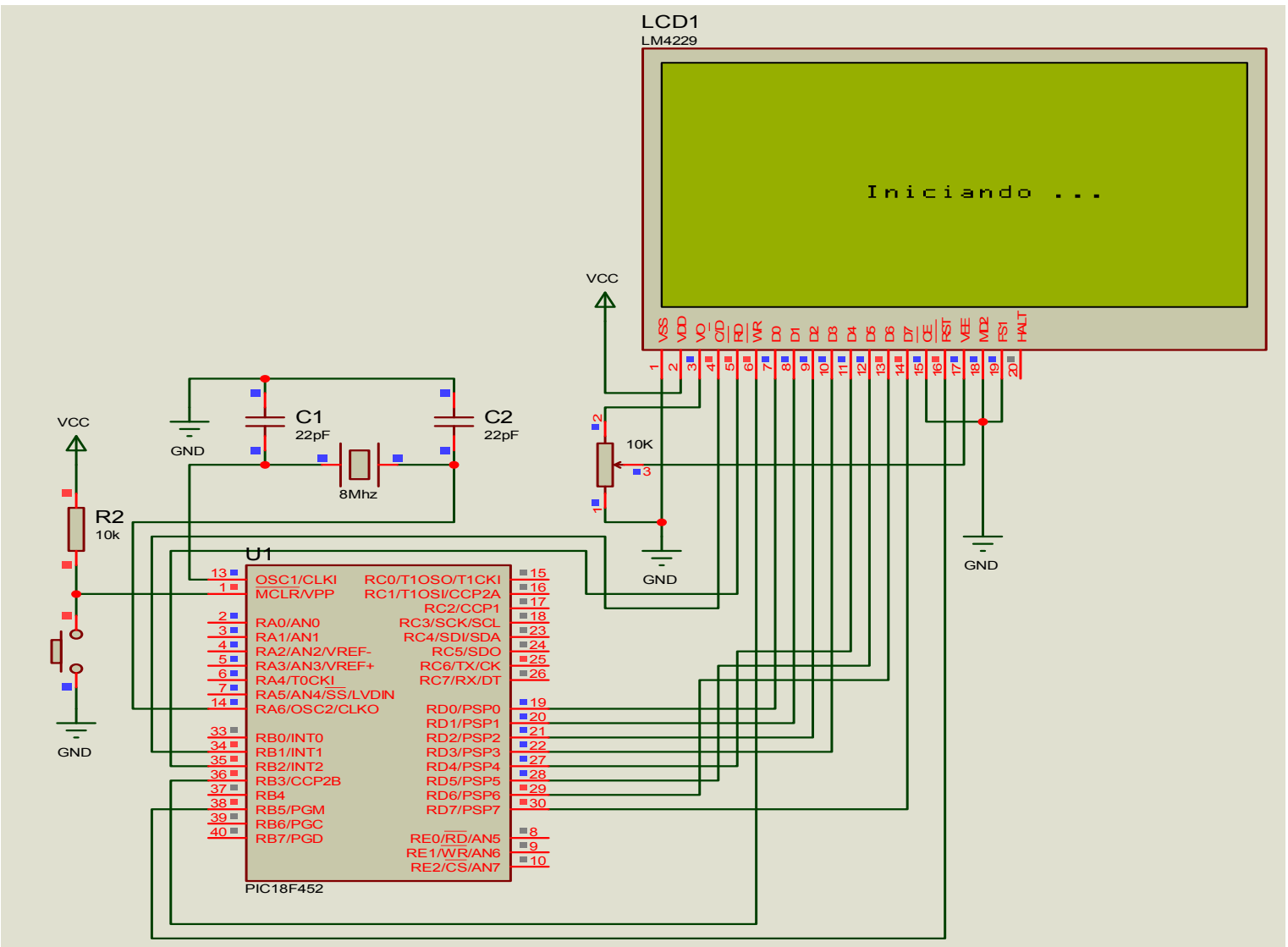


Figura 136: Diagrama de Simulación Módulo GLCD

En la siguiente figura, se muestra el diagrama circuital del Módulo GLCD, la conexión entre el microcontrolador y el panel resistivo táctil se realiza directamente usando 4 resistencias de 100  $\Omega$ , los datos interpretados son analógicos y dan la posición en X y Y.

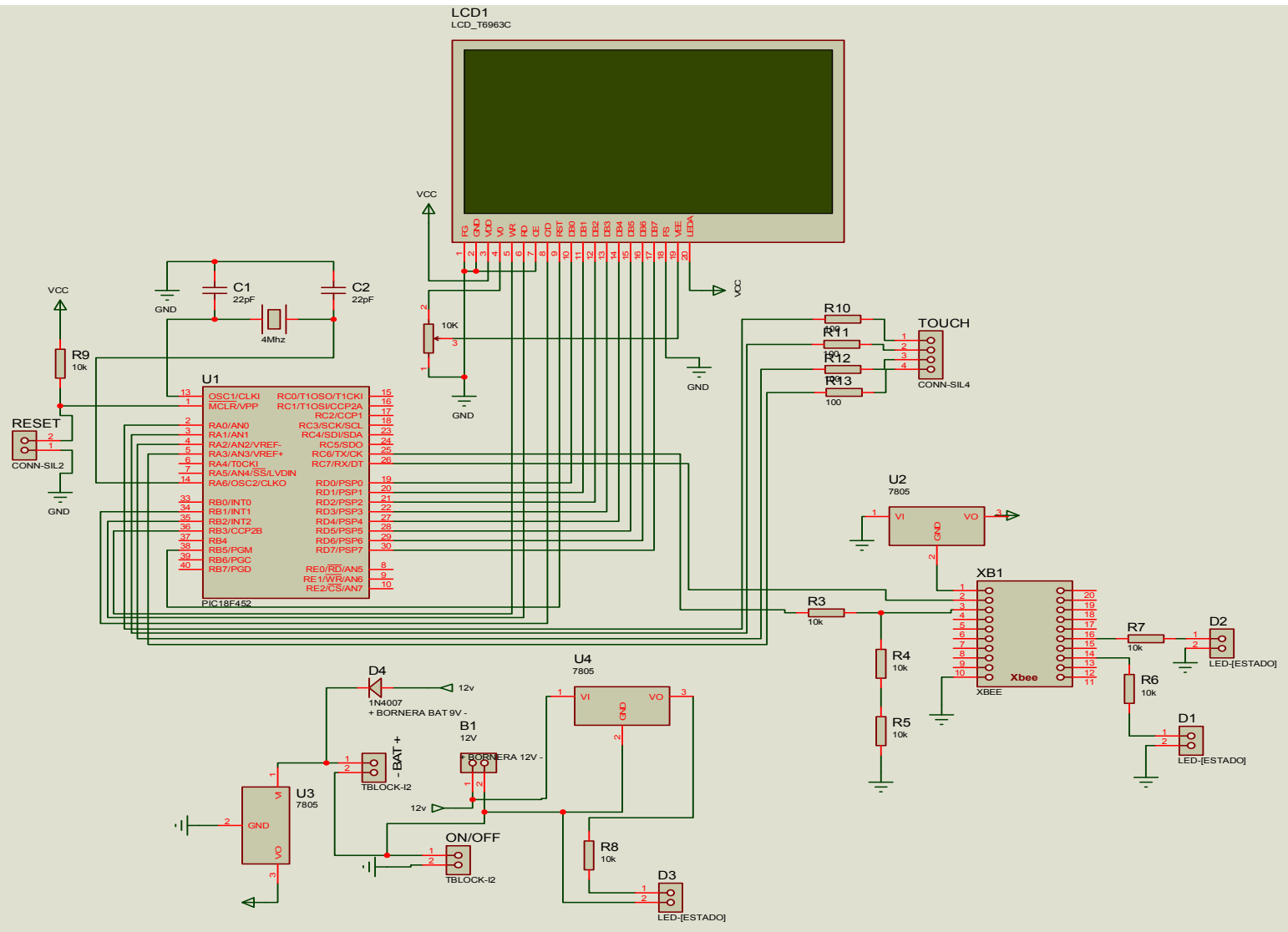


Figura 137: Diagrama Circuitual Módulo GLCD

En Proteus no existe el componente Xbee, se debe crear con la ayuda del datasheet, usando sus medidas y número de pines.

En la siguiente figura, se muestra el módulo GLCD enrutado en ARES, para la conexión del GLCD y su panel resistivo se usan buses de datos de 20 y 4 líneas.

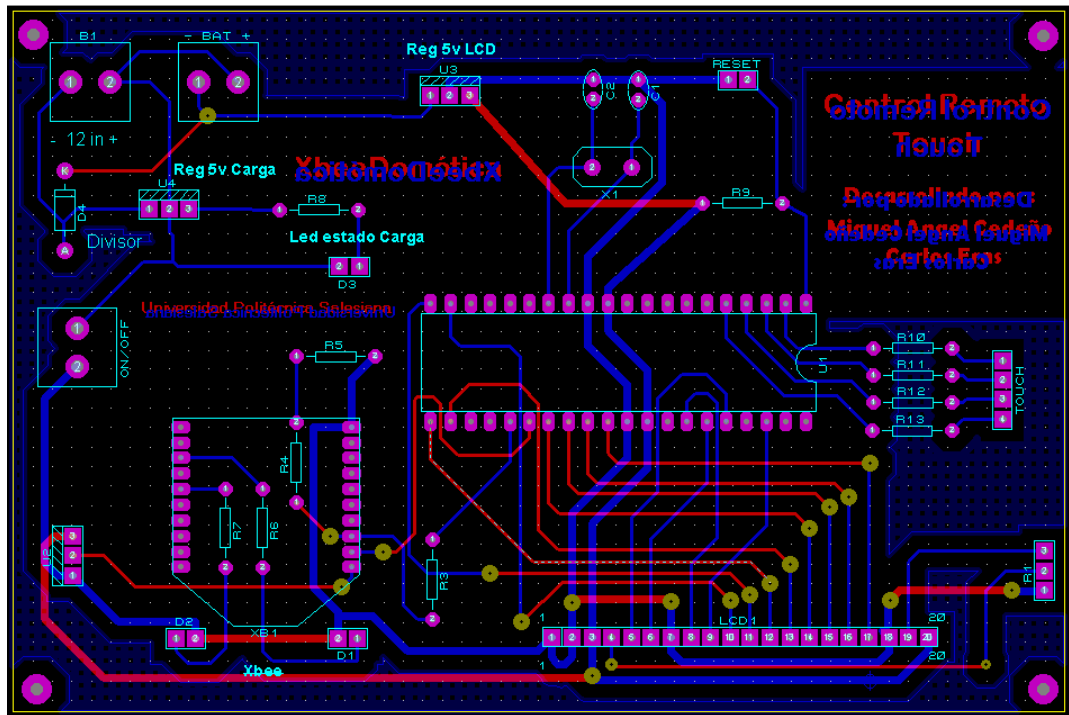


Figura 138: Circuito Módulo GLCD enrutado en ARES

En las siguientes figuras, se muestra la vista previa del circuito en 3D, de esta manera se tiene una mejor idea de cómo quedara distribuido físicamente.

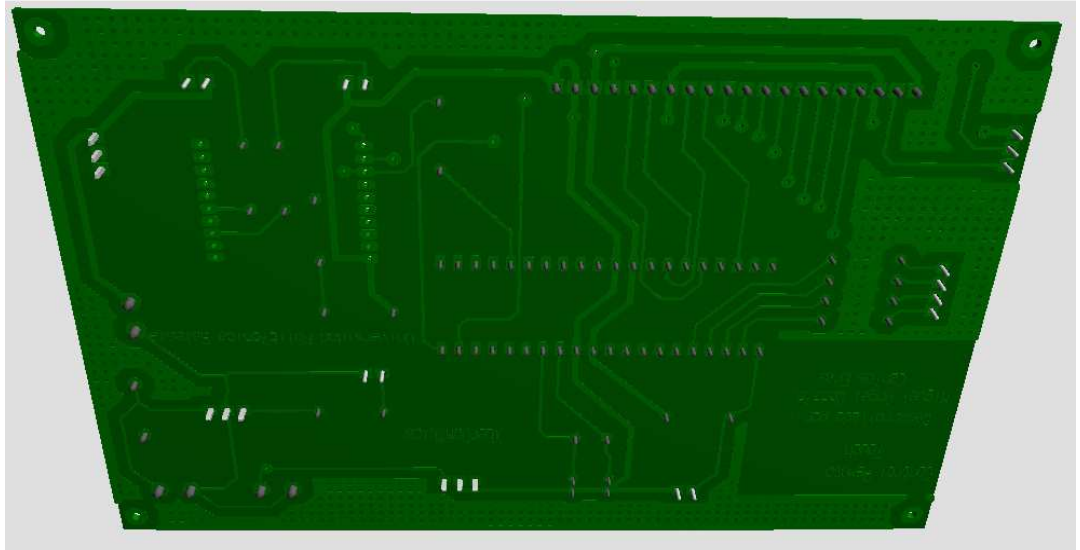


Figura 139: Vista 3D inferior Módulo GLCD

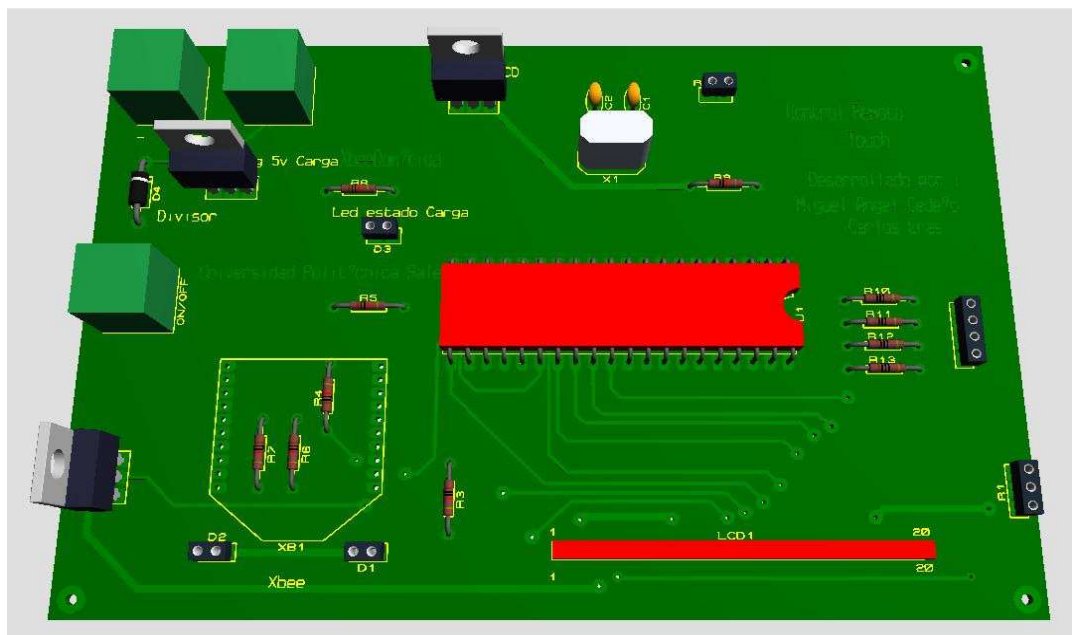


Figura 140: Vista 3D superior Módulo GLCD

Una vez diseñado y perfectamente enrutado el circuito en Ares, se procede a la impresión de las capas superiores e inferiores en este caso la figura 141 muestra la impresión de la capa cobre inferior, la figura 142 muestra la impresión de la capa cobre superior y la figura 143 muestra la impresión de la capa seda superior que contiene los nombres de los componentes, las capas superiores deben ser impresas en mirror.

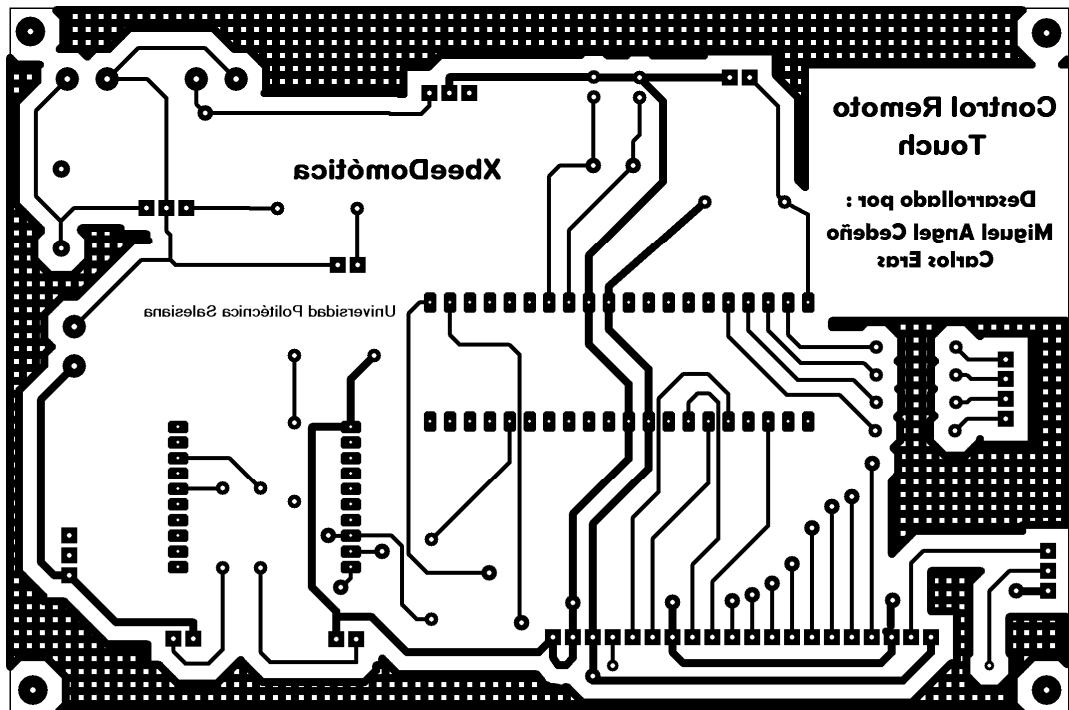


Figura 141: Bottom copper (Cobre inferior) Circuito Módulo GLCD

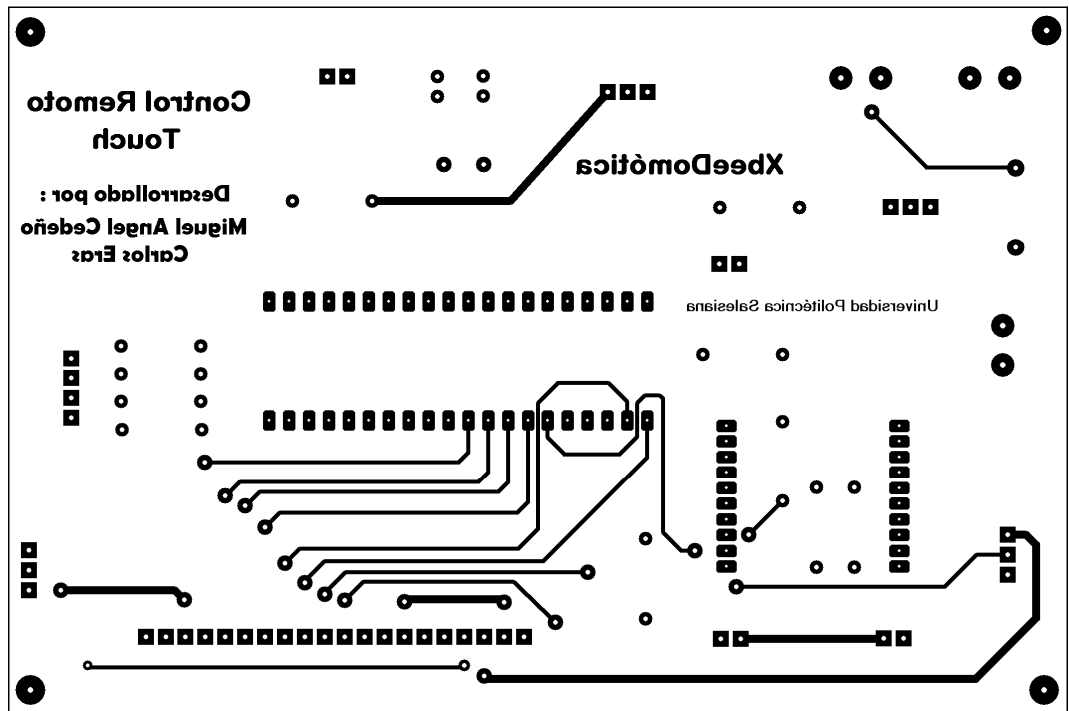
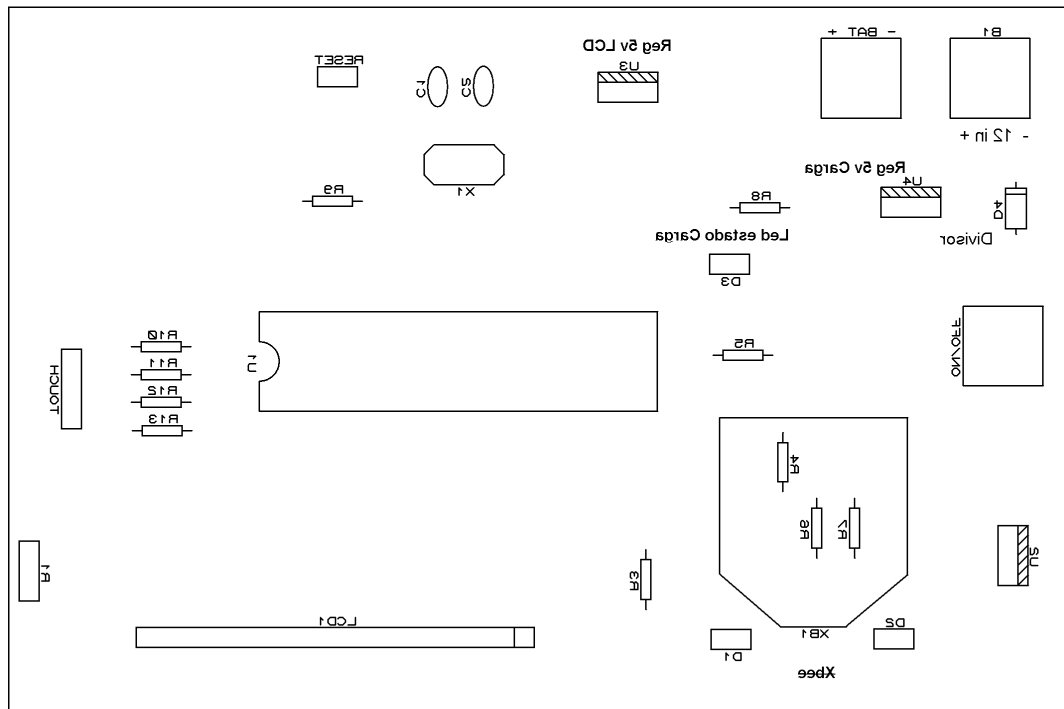
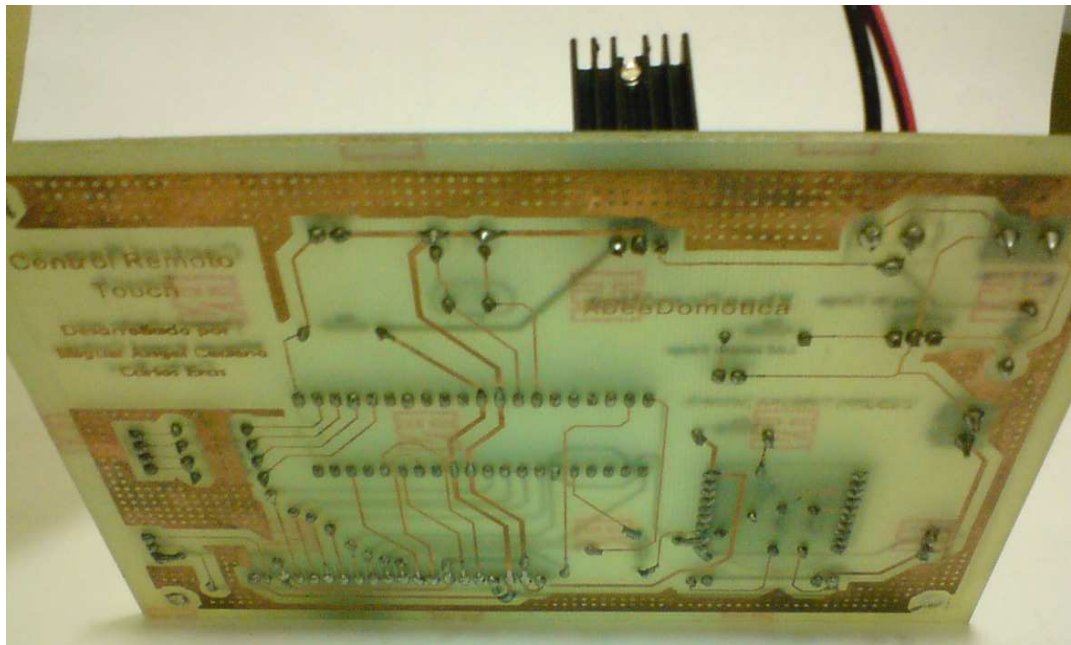


Figura 142: Top copper (Cobre superior) Circuito Módulo GLCD



En las siguientes figuras, se muestran las fotografías del circuito Módulo GLCD terminado.



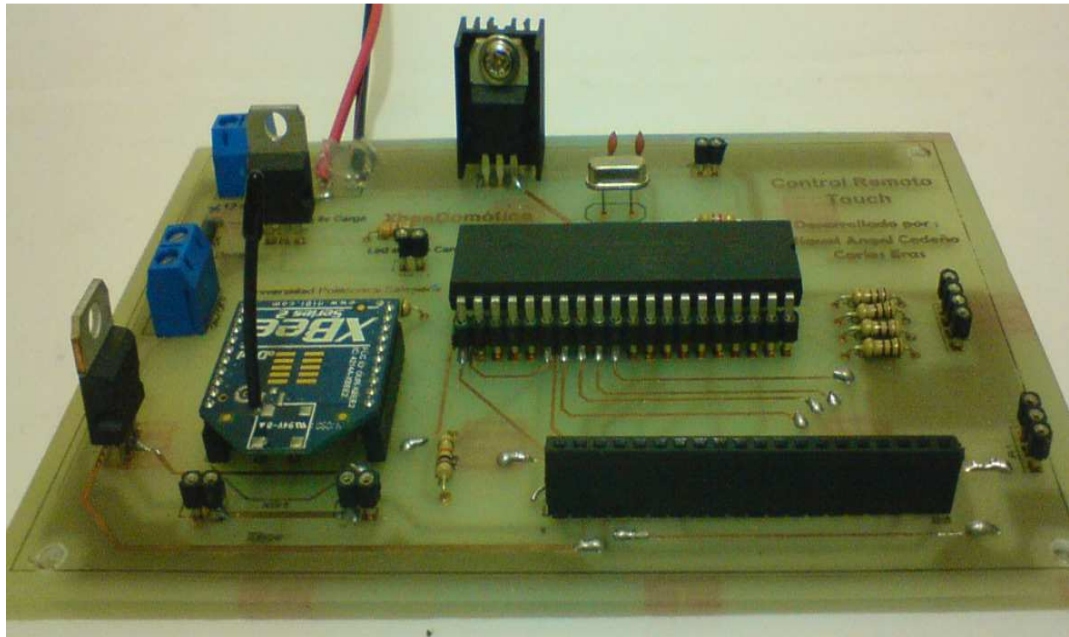


Figura 145: Fotografía superior del circuito Módulo GLCD

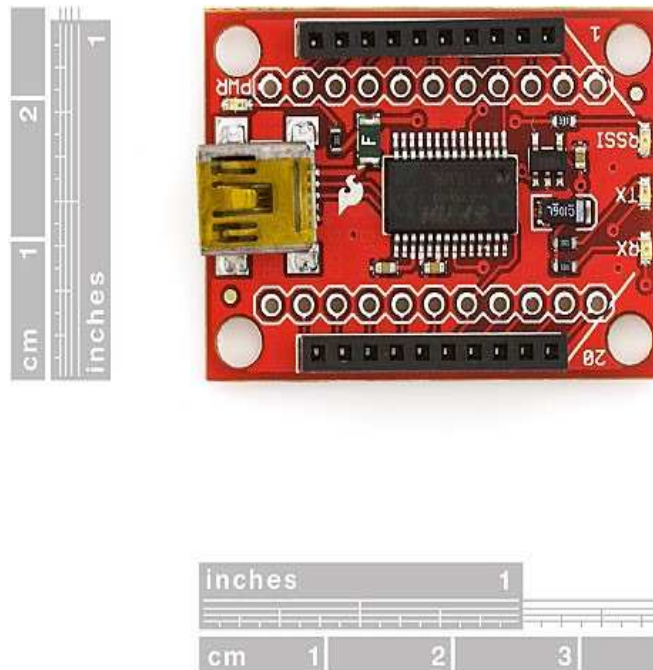
#### 5.2.6. Módulo Coordinador

Para la implementación del módulo coordinador se usa la interfaz Xbee Serial Explorer V12 USB, en la siguiente figura, se muestra el diagrama circuital de la interfaz USB.

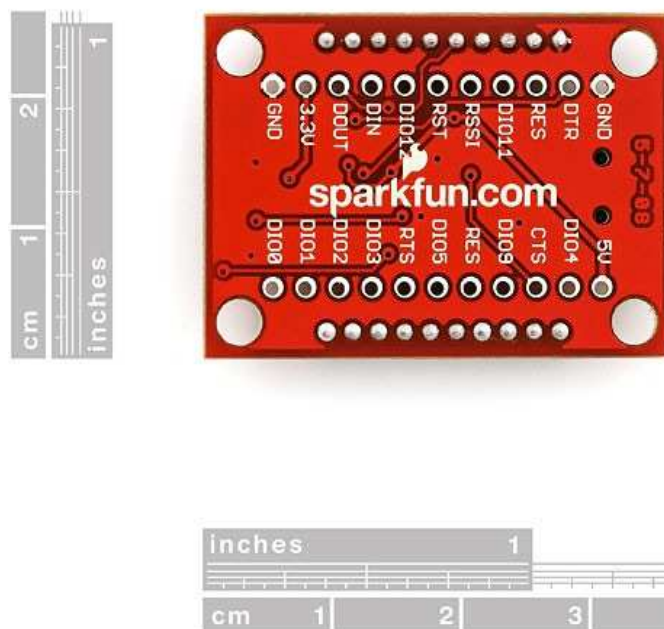




En la figura 147 se muestra la vista superior de la interfaz Xbee Serial Explorer V12 USB con las respectivas medidas en centímetros y en la figura 148 la vista inferior.

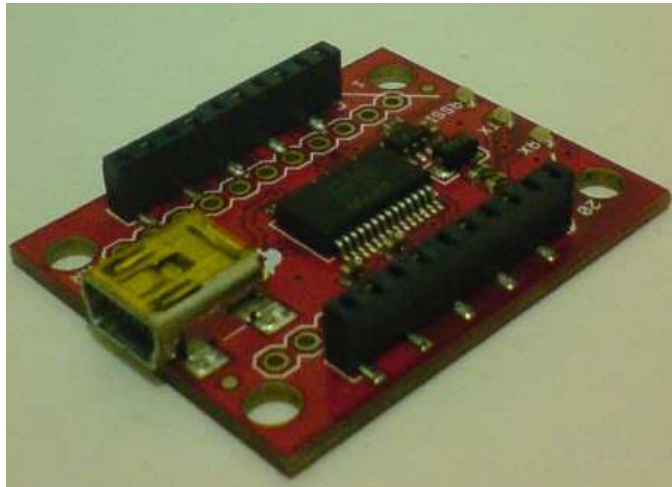


**Figura 147: Vista superior Xbee Serial Explorer V12 USB**  
Fuente: <http://www.bricogeek.com/shop/156-xbee-explorer-usb.html>

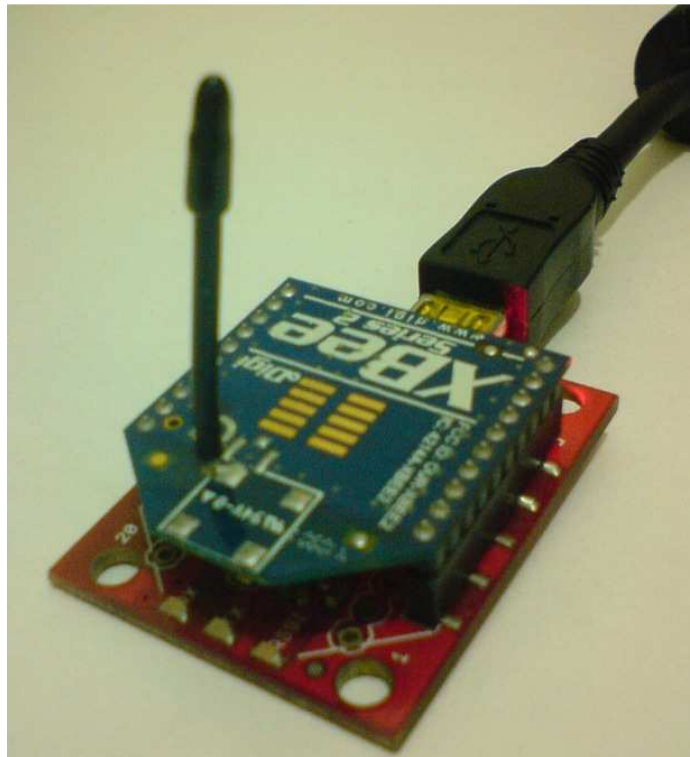


**Figura 148: Vista inferior Xbee Serial Explorer V12 USB**  
Fuente: <http://www.bricogeek.com/shop/156-xbee-explorer-usb.html>

En la Figura 149 se muestra la fotografía de la interfaz Serial Explorer y en la Figura 150 se muestra la fotografía de la interfaz con un módulo Xbee series 2.



**Figura 149: Fotografía del Xbee Serial Explorer V12 USB**



**Figura 150: Fotografía del Xbee Serial Explorer V12 USB con el módulo Xbee series 2**

## **5.3. DESARROLLO DE SOFTWARE**

El desarrollo de software implica la programación (generación de código) de cada uno de los componentes programables y aplicaciones del sistema.

### **5.3.1. MICROCONTROLADORES**

En todos los módulos se trabaja con el microcontrolador 18F452 por las especificaciones descritas en capítulos anteriores.

#### **5.3.1.1. Módulo Xbee1**

La programación de este módulo se realiza usando el desarrollador MicroCode Studio.

Se usan dos tipos de variables BYTE Y WORD.

Agregar la librería modedefs.bas para usar los controles de transmisión serial “serin” y “serout”.

```
include "modedefs.bas"
```

Escuchar el PORTC.7 en espera de un dato, si llega almacenarlo en la variable DATO de tipo WORD, si no llega en los próximos 50 milisegundos ir a la función PULSADORES.

```
serin PORTC.7,T9600,50,PULSADORES,DATO
```

Si la variable dato es igual a una de estas opciones hacer lo que se pide.

```
'SALA
IF dato = "A" THEN
HIGH PORTD.0
GOTO PULSADOR1:
endif
```

```
'BOCINA
IF dato = "H" THEN
HIGH PORTD.1
```

```
GOTO PULSADOR2:
```

```
endif
```

```
'CORTINABRIR
```

```
IF dato = "I" and PORTD.5 = 0 THEN
```

```
HIGH PORTD.2
```

```
GOTO CORTINABRIR:
```

```
endif
```

```
'CORTINACERRAR
```

```
IF dato = "J" and PORTD.6 = 0 THEN
```

```
HIGH PORTD.3
```

```
GOTO CORTINACERRAR:
```

```
Endif
```

Si el dato recibido vía serial es el carácter 'x' el microcontrolador devuelve el estado de cada uno de los sensores y actuadores conectados a él, para enviar el valor de las variables de estado que son de tipo WORD se usa el '#', para separar cada estado enviado es necesario un ENTER para esto se usa el número '13' y un tiempo entre envío como se indica en el siguiente código.

```
'ESTADO
```

```
IF dato = "x" THEN
```

```
SEROUT PORTC.6,T9600,[#Esala,13]
```

```
PAUSE 500
```

```
SEROUT PORTC.6,T9600,[#Ebocina,13]
```

```
PAUSE 500
```

```
SEROUT PORTC.6,T9600,[#Esenfoto,13]
```

```
PAUSE 500
```

```
SEROUT PORTC.6,T9600,[#Emov1,13]
```

```
PAUSE 500
```

```
SEROUT PORTC.6,T9600,[#Edethumo,13]
```

```
PAUSE 500
```

```
SEROUT PORTC.6,T9600,[#Ecort,13]
```

```
PAUSE 500
```

```
if Exbee = 0 then
```

```
SEROUT PORTC.6,T9600,["CorrectoX1",13]
```

```
ENDIF
```

```

    IF eXBEE = 1 THEN
    SEROUT PORTC.6,T9600,["FallaX1",13]
    ENDIF
PAUSE 500
Endif

```

Para el sensor Fotovoltaico se usa una variable analógica de tipo BYTE se obtienen los datos analógicos con el método "adcin" que van desde 0 hasta 255, luego es cuestión de compararlos para determinar la cantidad de luz que representa la noche y el día que es lo que se necesita saber. Con una prueba se verificó que cuando el sensor marca 50 las lámparas del alumbrado público se comienzan a encender, entonces se tomó el número 50 registrado como referencia.

```

var_analogica VAR byte
adcin 0,var_analogica
' Sensor fotovoltaico(analógico) LDR
  if var_analogica > 50 then
    'portb=%00000100
  ' Es de dia
    if CM3<1 then
      SEROUT PORTC.6,T9600,["1710",13]
      Esenfoto = 1710;
      cm4=0
      LOW PORTB.3
    endif
    cm3=1
  endif
  if var_analogica <= 50 then
    if CM4<1 then
      'portb=%000001000
    ' Es de noche
      SEROUT PORTC.6,T9600,["1711",13]
      Esenfoto = 1711;
      cm3=0
      HIGH PORTB.3
    cm4=1
  endif

```

El Detector de movimiento presenta un problema, se activa y se desactiva automáticamente cada vez que registra o no registra movimiento, si no se controla esto el sistema se puede ver saturado por tanta información enviada vía serial, la solución es poner una bandera y un contador de tiempo, que envíe una notificación de estado cuando fue activado el detector y comience la cuenta de un determinado tiempo para enviar la notificación de desactivado, si el detector se desactiva y se vuelve activar antes que la cuenta termine, el contador de tiempo se reinicia con el fin de enviar menos notificaciones, solo cuando sean necesarias.

```
Detector_mov1:
  IF PORTC.2 = 0 THEN
    MOV=0
    if CONT4=0 THEN
      SEROUT PORTC.6,T9600,["1211",13]
      Emov1=1211
      CT = 0
      CONT4=1
      CONT3=0
    ENDIF
  endif
  IF PORTC.2 = 1 and CT = 0 THEN
    Emov1=1210
    MOV=0
    CT = 1
  endif
  if MOV >200 AND CONT3=0 THEN
    SEROUT PORTC.6,T9600,["1210",13]
    CT=0
    CONT4=0
    CONT3=1
  ENDIF
  Contador_tiempo:
  if CT = 1 THEN
    MOV = MOV + 1
  ENDIF
```

El detector de humo funciona igual que el detector de movimiento hay que tener en cuenta las mismas recomendaciones para solucionar el problema de envío excesivo de notificaciones.

En el control de los pulsadores el problema es que se desconoce cuánto se demora un usuario al presionar un botón, si mantiene presionado el pulsador por mucho tiempo el sistema puede enviar n número de veces la misma notificación, para solucionar este problema se usan banderas como se muestra en el siguiente fragmento de código.

PULSADOR1:

```
IF PORTC.0 = 1 THEN
  LOW PORTD.0 ; RECIBIDO
  while PORTC.0=1
  wend
  cont=cont+1
  if cont > 1 then
    cont=0
  endif
endif
select case cont
  case 0
    low PORTB.0
    low PORTB.4
  case 1
    HIGH PORTB.0
    HIGH PORTB.4
```

Para el control de apertura y cierre de la cortina se usan pulsadores y sensores magnéticos que funcionen como fin de ciclo, también es importante notar que la cortina no puede recibir órdenes de apertura y cierre a la vez, por lo que el uso de banderas entre abrir y cerrar se hace indispensable como se muestra siguiente fragmento de código.

CORTINACERRAR:

```
IF PORTC.4 = 1 AND PORTD.6= 0 THEN
```



```

LOW PORTB.7
LOW PORTA.2
cont8=0
LOW PORTD.2 ; RECIBIDO
    while PORTC.4=1
    wend
    cont7=cont7+1
    if cont7 > 1 then
        cont7=0
    endif
select case cont7
case 0
    LOW PORTB.6
    LOW PORTA.1
case 1
    HIGH PORTB.6
    HIGH PORTA.1
end select
endif

```

CORTINAABRIR:

```

IF PORTC.5 = 1 AND PORTD.7= 0 THEN
LOW PORTB.6
LOW PORTA.1
cont7=0
LOW PORTD.3 ; RECIBIDO
    while PORTC.5=1
    wend
    cont8=cont8+1
    if cont8 > 1 then
        cont8=0
    endif
select case cont8
case 0
    LOW PORTB.7
    LOW PORTA.2
case 1
    HIGH PORTB.7
    HIGH PORTA.2
end select

```

El detector de energía informa cuando el sistema sufrió un corte de energía, funciona similar a un pulsador, el puerto del microcontrolador recibe una entrada digital y este la interpreta como falla de suministro eléctrico o como restablecimiento del suministro.

El monitoreo de los diferentes sensores y actuadores se realiza cada 50 milisegundos porque el programa al llegar al final vuelve al inicio, escucha su puerto serial por 50 milisegundos y continúa con el resto del programa, así se repite el ciclo una y otra vez.

Para mayor información sobre la programación del microcontrolador ubicado en el módulo Xbee 1 ver el código en el anexo digital CD-ROM:\CIRCUITOS\Xbee1\xbee1.pbp

#### **5.3.1.2. Módulo Xbee2**

La programación de este módulo se realiza usando el desarrollador MicroCode Studio, es muy similar a la desarrollada en el módulo Xbee 1.

Se usan dos tipos de variables BYTE Y WORD.

Agregar la librería modedefs.bas para usar los controles de transmisión serial "serin" y "serout".

```
include "modedefs.bas"
```

Escuchar el PORTC.7 en espera de un dato si llega almacenarlo en la variable DATO de tipo WORD, si no llega en los próximos 50 milisegundos ir a la función PULSADORES.

```
serin PORTC.7,T9600,50,PULSADORES,DATO
```

Si la variable dato es igual a una de estas opciones hacer lo que se pide.

'DORMITORIO

```
IF dato = "D" THEN
HIGH PORTD.0
GOTO PULSADOR1:
endif
```

'COCINA

```
IF dato = "C" THEN
HIGH PORTD.1
GOTO PULSADOR2:
Endif
```

Si el dato recibido vía serial es el carácter 'y' el microcontrolador devuelve el estado de cada uno de los sensores y actuadores conectados a él, para enviar el valor de las variables de estado que son de tipo WORD se usa el '#', para separar cada estado enviado es necesario un ENTER para esto se usa el número '13' y un tiempo entre envíos como se indica en el siguiente código.

'ESTADO

```
IF dato = "y" THEN
SEROUT PORTC.6,T9600,[#Edormi,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Ecocina,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Edetecgas,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Emov2,13]
PAUSE 500
    if Exbee = 0 then
        SEROUT PORTC.6,T9600,["CorrectoX2",13]
        ENDIF
    IF eXBEE = 1 THEN
        SEROUT PORTC.6,T9600,["FallaX2",13]
        ENDIF
PAUSE 500
Endif
```

Para el detector de gas GLP se usa una variable analógica de tipo BYTE se obtienen los datos analógicos con el método “adcin” que van desde 0 hasta 255, luego se compara para determinar la cantidad de gas que está presente en una habitación. Con una prueba se verificó que cuando el sensor marca menos de 6 la presencia de gas GLP en la habitación es mínima y no representa peligro alguno cuando sube este nivel es decir, entre 6 – 255 el detector se activa y el microcontrolador envía una alerta al sistema y activa una chicharra.

adcin 0,dog 'Sensor de gas MG-6 conectado a pin 0 del Puerto A (analógico)

```

if dog<6 then
  if CM3<1 then
    SEROUT PORTC.6,T9600,["2410",13]
    Edetecgas =2410
    cm4=0
    HIGH PORTB.2
    LOW PORTB.3
    LOW PORTD.4
  endif
endif
if dog>6 then
  if CM4<1 then
    SEROUT PORTC.6,T9600,["2411",13]
    Edetecgas =2411
    cm3=0
    HIGH PORTD.4 ' CHICHARRA
    HIGH PORTB.3
    LOW PORTB.2
  endif
cm4=1
Endif

```

El detector de movimiento y el detector de energía ubicado en este módulo es idéntico al que se encuentra en el módulo Xbee1 por lo tanto el código es el mismo.

En el control de los pulsadores el problema es que se desconoce cuánto se demora un usuario al presionar un botón, si mantiene

presionado el pulsador por mucho tiempo el sistema puede enviar n número de veces la misma notificación, para solucionar este problema se usan banderas al igual que en el desarrollo del código del módulo Xbee 1.

El monitoreo de los diferentes sensores y actuadores se realiza cada 50 milisegundos porque el programa al llegar al final vuelve al inicio, escucha su puerto serial por 50 milisegundos y continúa con el resto del programa, así se repite el ciclo una y otra vez.

Para mayor información sobre la programación del microcontrolador ubicado en el módulo Xbee 2 ver el código en el anexo digital CD-ROM:\CIRCUITOS\Xbee2\xbee2.pbp .

#### **5.3.1.3. Módulo Xbee3**

La programación de este módulo se realiza usando el desarrollador MicroCode Studio, es muy similar a la desarrollada en los módulos Xbee 1 y Xbee 2.

Se usan dos tipos de variables BYTE Y WORD.

Agregar la librería modedefs.bas para usar los controles de transmisión serial “serin” y “serout”.

```
include "modedefs.bas"
```

Escuchar el PORTC.7 en espera de un dato si llega almacenarlo en la variable DATO de tipo WORD, si no llega en los próximos 50 milisegundos ir a la función PULSADORES.

```
serin PORTC.7,T9600,50,PULSADORES,DATO
```

Si la variable dato es igual a una de estas opciones hacer lo que se pide.

```
'EXTERIOR
IF dato = "B" THEN
HIGH PORTD.0
GOTO PULSADOR1:
endif
```

```
'CERRADURA1
IF dato = "F" THEN
HIGH PORTD.1
GOTO PULSADOR2:
endif
```

```
'CERRADURA2
IF dato = "G" THEN
HIGH PORTD.2
GOTO PULSADOR3:
endif:
```

Si el dato recibido vía serial es el carácter 'z' el microcontrolador devuelve el estado de cada uno de los sensores y actuadores conectados a él, para enviar el valor de las variables de estado que son de tipo WORD se usa el '#' , para separar cada estado enviado es necesario un ENTER para esto se usa el número '13' y un tiempo entre envió como se indica en el siguiente código.

```
'ESTADO
IF dato = "z" THEN
SEROUT PORTC.6,T9600,[#Eext,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Esenmag1,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Esenmag2,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Ecerradura1,13]
PAUSE 500
SEROUT PORTC.6,T9600,[#Ecerradura2,13]
PAUSE 500
```

```

if Exbee = 0 then
SEROUT PORTC.6,T9600,["CorrectoX3",13]
ENDIF
IF eXBEE = 1 THEN
SEROUT PORTC.6,T9600,["FallaX3",13]
ENDIF
PAUSE 500
Endif

```

En el control de los pulsadores el problema es que se desconoce cuánto se demora un usuario al presionar un botón, si mantiene presionado el pulsador por mucho tiempo el sistema puede enviar n número de veces la misma notificación, para solucionar este problema se usan banderas al igual que en el desarrollo del código del módulo Xbee 1 y Xbee 2.

Las cerraduras eléctricas funcionan con pulsos eléctricos que activan una bobina y esta abre el cerrojo, por lo tanto en la programación se debe tomar en cuenta esto, la cerradura no puede estar activa por mucho tiempo así que un segundo es suficiente para abrirla, luego de esto se desactiva, en el siguiente fragmento se muestra la programación de la cerradura 1.

```

'CERRADURA 1
PULSADOR2:
IF PORTC.1 = 1 THEN
LOW PORTD.1 ; RECIBIDO
while PORTC.1=1
wend
cont2=cont2+1
if cont2 > 1 then
cont2=0
endif
endif
select case cont2
case 0
low PORTB.6
low PORTB.4
case 1

```

```

HIGH PORTB.6
HIGH PORTB.4
pause 1000
cont2=0
end select

```

Los sensores magnéticos informan si una puerta está abierta o cerrada enviando una señal digital hacia el microcontrolador, funcionan igual que un pulsador normalmente cerrado, cuando la puerta está abierta envían una señal baja y cuando la puerta está cerrada envían una señal alta.

' SENSOR MAGNETICO 1

Sensor\_Mag1:

```

IF PORTC.3 = 0 THEN
  if cont4<1 THEN
    SEROUT PORTC.6,T9600,["3511",13]
    HIGH PORTB.1
    Esenmag1=3511
    cont5=0
  endif
  cont4=1
endif
IF PORTC.3 = 1 THEN
  if cont5<1 THEN
    SEROUT PORTC.6,T9600,["3510",13]
    LOW PORTB.1
    Esenmag1=3510
    cont4=0
  endif
  cont5=1
endif

```



El monitoreo de los diferentes sensores y actuadores se realiza cada 50 milisegundos porque el programa al llegar al final vuelve al inicio, escucha su puerto serial por 50 milisegundos y continúa con el resto del programa, así se repite el ciclo una y otra vez.

Para mayor información sobre la programación del microcontrolador ubicado en el módulo Xbee 3 ver el código en el anexo digital CD-ROM:\CIRCUITOS\Xbee3\xbee3.pbp

#### **5.3.1.4. Módulo GLCD Touch**

La programación de este módulo se realiza usando el desarrollador MikroBasic.

Se usan dos tipos de variables BYTE, STRING Y WORD.

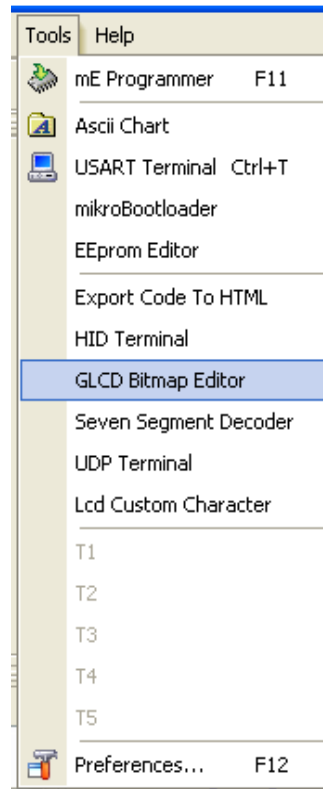
Agregar "T6963c" para usar los métodos de control del GLCD, "imagenes" para usar las imagenes hechas vector y "subfunciones" como si se tratara de una clase, llamar a las funciones que estén en ella.

```
include "T6963c"  
include "imagenes"  
include "subfunciones"
```

#### **Módulo imagenes.pbas**

Este módulo se compone por las imágenes codificadas en un vector que se van a presentar en el GLCD.

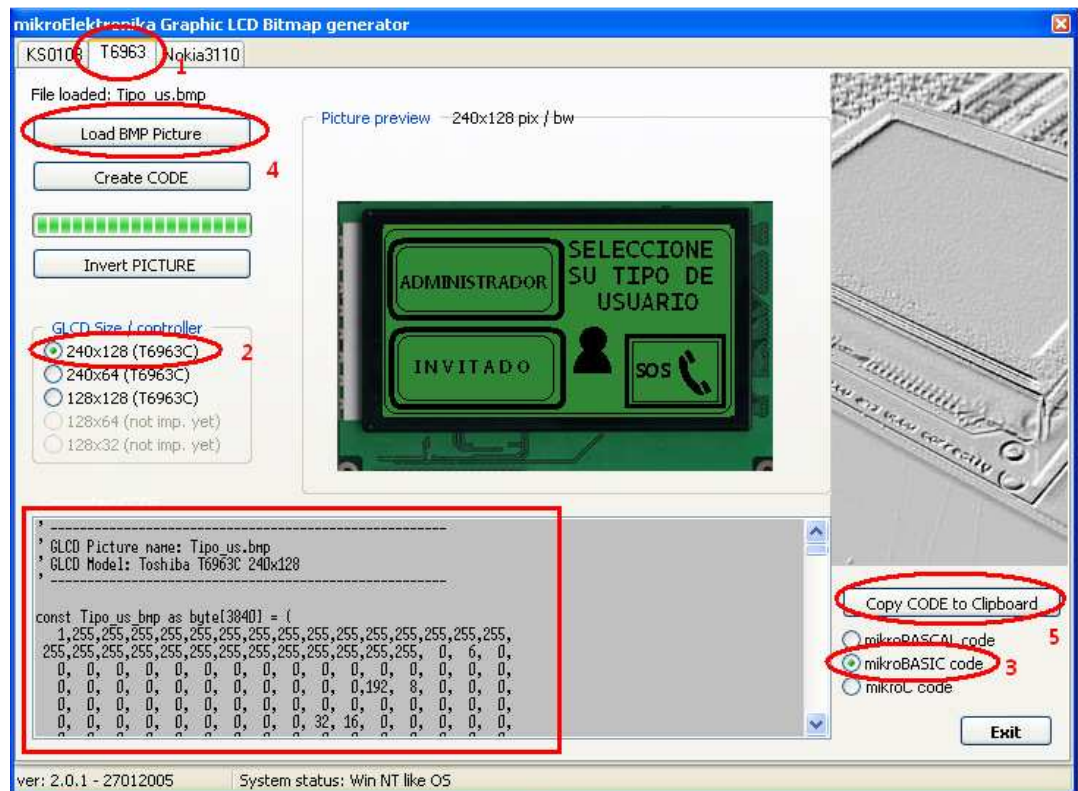
La manera de obtener el vector de una imagen es usando la herramienta de MikroBasic "GLCD Bitmap Editor" ubicada en tools como se muestra en la siguiente figura.



**Figura 151: Pasos para abrir GLCD Bitmap editor**

Fuente: MikroBasic versión 7.0.0.2

Para codificar una imagen bmp en un vector se debe primero seleccionar el tipo de GLCD que se está usando en este caso el T6963, en segundo lugar la resolución o el tamaño del GLCD en este caso 240 x 128, en tercer lugar el lenguaje al cual se va a importar este código en este caso se selecciona MikroBasic, en cuarto lugar se carga la imagen con load BMP Picture hay que tener en cuenta que la imagen que se seleccione debe ser .bmp, monocromática y tener una resolución de 240 x 128, de manera automática se genera el código de esta imagen, como último paso se copia y listo, en la siguiente figura, se muestra paso a paso este proceso.



**Figura 152: GLCD Bitmap generator**

Fuente: MikroBasic versión 7.0.0.2

La estructura del módulo imágenes viene dado de la siguiente manera

module imágenes

Código generado por la imagen Tipo\_us.bmp

Código generado por la imagen clave\_ad.bmp

Código generado por la imagen luces.bmp

Código generado por la imagen sensores.bmp

Código generado por la imagen cerraduras.bmp

implements

end.

## Módulo subfunciones.pbas

Inicializa el GLCD , asigna los puertos de bus y de control.

```
sub procedure Initialize
    ADCON0 = 2
    TRISA = 0
    PORTA = 0
    LineFlag = False
    T6963C_init(240, 128, 8, PORTD, PORTB, 3, 2, 1, 5)
    Usart_Init(9600)
    T6963C_graphics(1)
    delay_ms(100)
    T6963C_text(1)
end sub
```

Para obtener las coordenadas en los ejes “X” y “Y” se usan estas subfunciones que comparan valores analógicos recibidos y devuelven el valor digital exacto de donde fue presionada la pantalla táctil.

```
sub function Ypos as word
    dim tmp as word
    TRISA = (TRISA And 240) Or 4      ' AN2 como entrada
    PORTA = 10                       ' energiza X , RA1 y RA3
    result = ADC_Read(2)              ' obtiene valor de eje Y
    If result <> 0 Then
        TRISA.2 = 0
        TRISA.0 = 1
        tmp = (1023 - ADC_Read(0))
        result = (result + tmp) >> 1
    End If
end sub
```

```
sub function Xpos as word
    dim tmp as word
    TRISA = (TRISA And 240) Or 2      ' AN1 como entrada
    PORTA = 5                         ' energiza Y , RA0 and RA2
    result = ADC_Read(1)              ' obtiene valor de eje X
    If result <> 0 Then
```

```

TRISA.1 = 0
TRISA.3 = 1
tmp = (1023 - ADC_Read(3))
result = (result + tmp) >> 1
End If
end sub

```

Una vez obtenidos los valores X y Y se imprime para saber en qué parte del panel resistivo táctil el usuario presionó.

```

sub procedure PrintXY
T6963C_WRITE_TEXT("X=",0,0,1)
T6963C_WRITE_TEXT("Y=",0,1,1)
'touch
x=xpos 'y
y=ypos 'x
wordtostr(xpos,mx)
T6963C_WRITE_TEXT(mx,3,0,1)
wordtostr(ypos,my)
T6963C_WRITE_TEXT(my,3,1,1)
end sub

```

Ahora se grafica cada una de las imágenes en el GLCD con el objetivo de hacer una matriz con las coordenadas de cada botón o de cada área, que se necesite registrar cuando el usuario presione el panel táctil.

```
T6963C_image(imagen_bmp)
```

En la siguiente figura, se muestra la imagen tipo de usuario con las respectivas coordenadas de cada botón, se debe controlar dos columnas y dos filas.

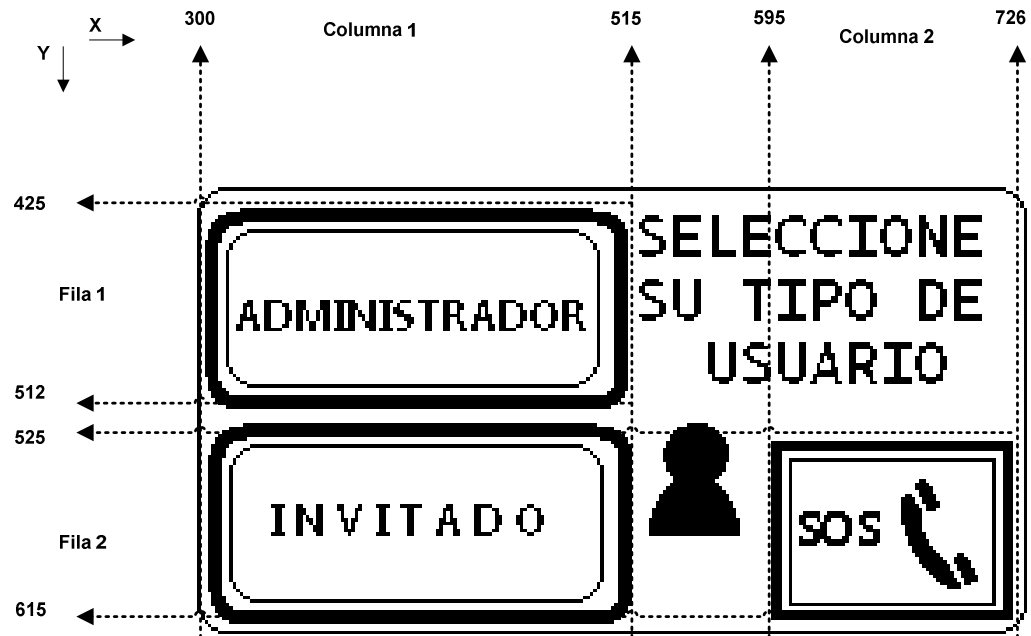


Figura 153: Coordenadas de la imagen Tipo de Usuario

El siguiente fragmento de código, muestra el sub proceso que identifica las columnas, filas y los botones o también denominados áreas de presión de la imagen tipo de usuario.

```
sub procedure Touch_tipous()
    x=xpos 'x
    y=ypos 'y

    'Columnas
    if (x >= 300) and (x < 515) then columna=1 end if
    if (x >= 595) and (x < 726) then columna=2 end if
    if xpos = 0 then fila=0 end if
    'Filas
    if (y >= 425) and (y < 512) then fila=1 end if
    if (y >= 525) and (y < 615) then fila=2 end if
    if ypos = 0 then columna=0 end if
    'botones (areas de presion)
    if (columna=1) and (fila=1) then dato=1 end if
    if (columna=1) and (fila=2) then dato=2 end if
    if (columna=2) and (fila=2) then dato=3 end if
    if (columna=0) and (fila=0) then dato=16 end if
end sub
```

En la siguiente figura, se muestra la imagen ingresar clave con las respectivas coordenadas de cada botón, se deben controlar 5 columnas y 3 filas.

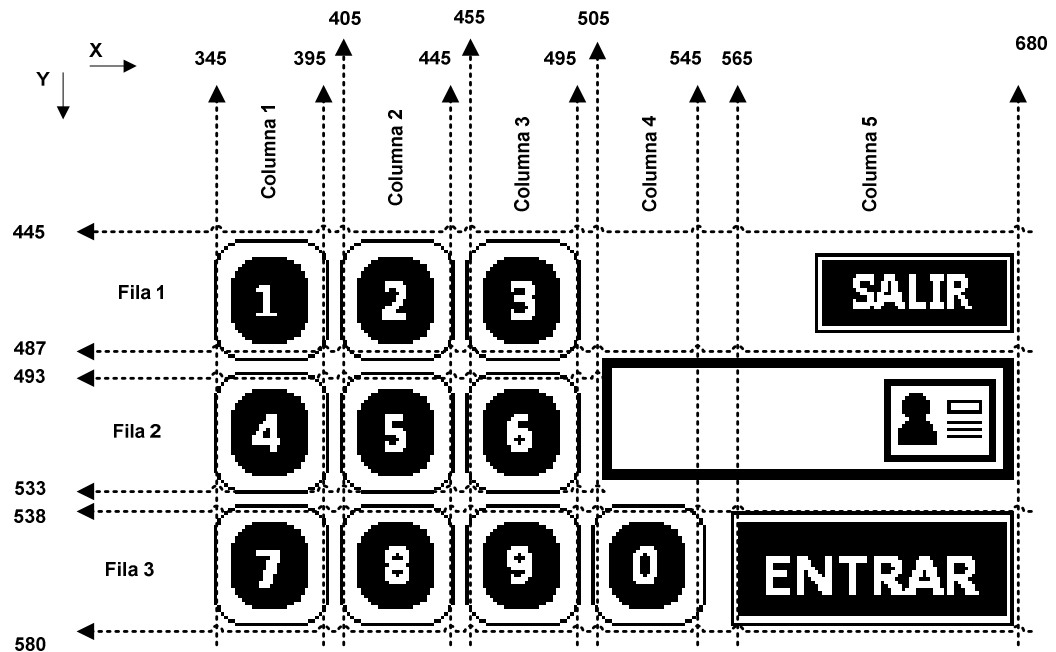


Figura 154: Coordenadas de la imagen Ingresar clave

El siguiente fragmento de código, muestra el sub proceso que identifica las columnas, filas y los botones o también denominados áreas de presión de la imagen Ingresar clave.

```
sub procedure Touch_clave()
    X=xpos 'x
    Y=ypos 'y
    'Columnas
    if (x >= 345) and (x < 395) then columna=1 end if
    if (x >= 405) and (x < 445) then columna=2 end if
    if (x >= 455) and (x < 495) then columna=3 end if
    if (x >= 505) and (x < 545) then columna=4 end if
    if (x >= 565) and (x < 680) then columna=5 end if
    if xpos = 0 then fila=0 end if
    'Filas
```

```

if (y >= 445) and (y < 487) then fila=1 end if
if (y >= 493) and (y < 533) then fila=2 end if
if (y >= 538) and (y < 580) then fila=3 end if
if ypos = 0 then columna=0 end if
'botones (areas de presion)
if (columna=1) and (fila=1) then dato=1 end if
if (columna=2) and (fila=1) then dato=2 end if
if (columna=3) and (fila=1) then dato=3 end if
if (columna=1) and (fila=2) then dato=4 end if
if (columna=2) and (fila=2) then dato=5 end if
if (columna=3) and (fila=2) then dato=6 end if
if (columna=1) and (fila=3) then dato=7 end if
if (columna=2) and (fila=3) then dato=8 end if
if (columna=3) and (fila=3) then dato=9 end if
if (columna=4) and (fila=3) then dato=0 end if
if (columna=5) and (fila=3) then dato=10 end if
if (columna=5) and (fila=1) then dato=11 end if
if (columna=0) and (fila=0) then dato=16 end if
end sub

```

En la siguiente figura, se muestra la imagen control de iluminación con las respectivas coordenadas de cada botón, se deben controlar 3 columnas y 4 filas.

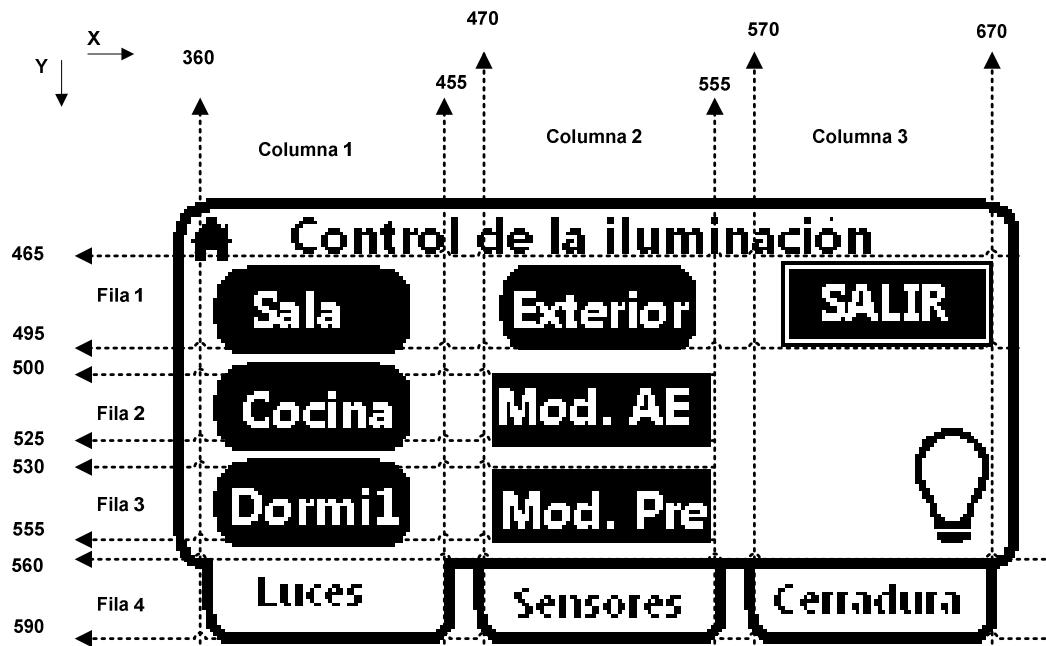


Figura 155: Coordenadas de la imagen control de la iluminación



El siguiente fragmento de código, muestra el sub proceso que identifica las columnas, filas y los botones o también denominados áreas de presión de la imagen control de iluminación.

```
sub procedure Touch_luces()
    x=xpos 'x
    y=ypos 'y
    'Columnas
    if (x >= 360) and (x < 455) then columna=1 end if
    if (x >= 470) and (x < 555) then columna=2 end if
    if (x >= 570) and (x < 670) then columna=3 end if
    if xpos = 0 then columna=0 end if
    'Filas
    if (y >= 465) and (y < 495) then fila=1 end if
    if (y >= 500) and (y < 525) then fila=2 end if
    if (y >= 530) and (y < 555) then fila=3 end if
    if (y >= 560) and (y < 590) then fila=4 end if
    if ypos = 0 then fila=0 end if
    'botones (areas de presion)
    if (columna=1) and (fila=1) then dato=1 end if
    if (columna=2) and (fila=1) then dato=2 end if
    if (columna=3) and (fila=1) then dato=3 end if
    if (columna=1) and (fila=2) then dato=4 end if
    if (columna=2) and (fila=2) then dato=5 end if
    if (columna=1) and (fila=3) then dato=6 end if
    if (columna=2) and (fila=3) then dato=7 end if
    if (columna=1) and (fila=4) then dato=8 end if
    if (columna=2) and (fila=4) then dato=9 end if
    if (columna=3) and (fila=4) then dato=10 end if
    if (columna=0) and (fila=0) then dato=16 end if
end sub
```

En la siguiente figura, se muestra la imagen monitoreo de sensores con las respectivas coordenadas de cada botón, se deben controlar 3 columnas y 1 fila.

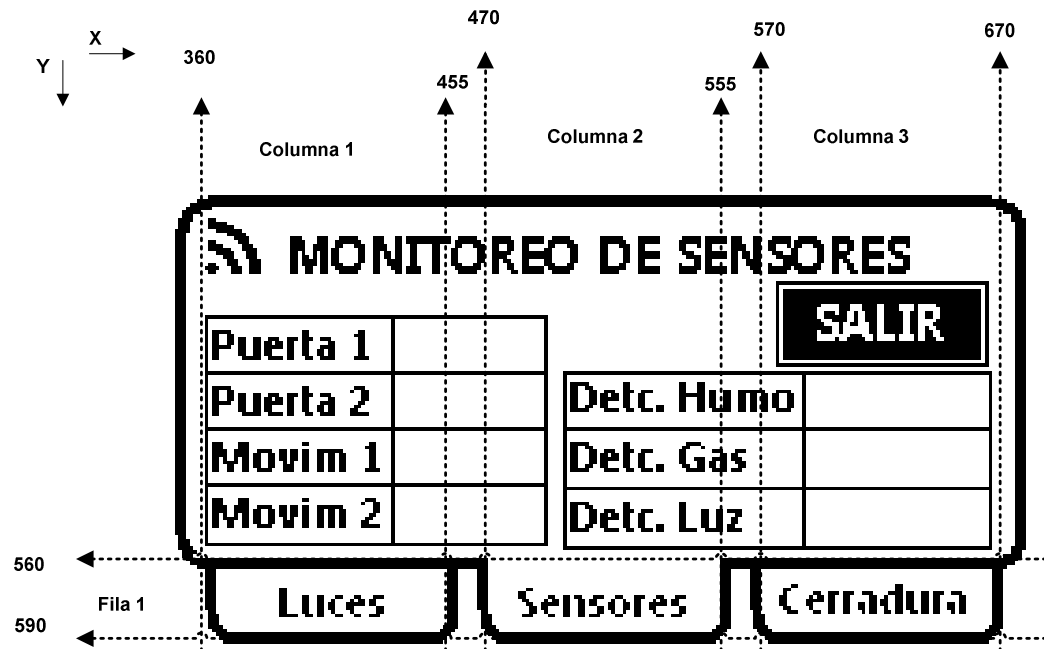


Figura 156: Coordenadas de la imagen Monitoreo de sensores

El siguiente fragmento de código, muestra el sub proceso que identifica las columnas, filas y los botones o también denominados áreas de presión de la imagen Monitoreo de sensores.

```
sub procedure Touch_sensores()
    x=xpos 'x
    y=ypos 'y
    'Columnas
    if (x >= 360) and (x < 455) then columna=1 end if
    if (x >= 470) and (x < 555) then columna=2 end if
    if (x >= 570) and (x < 670) then columna=3 end if
    if xpos = 0 then columna=0 end if
    'Filas
    if (y >= 470) and (y < 500) then fila=1 end if
    if (y >= 560) and (y < 590) then fila=2 end if
    if ypos = 0 then fila=0 end if
    'botones (areas de presion)
    if (columna=3) and (fila=1) then dato=1 end if
    if (columna=1) and (fila=2) then dato=2 end if
    if (columna=2) and (fila=2) then dato=3 end if
    if (columna=3) and (fila=2) then dato=4 end if
    if (columna=0) and (fila=0) then dato=16 end if
```

En la siguiente figura, se muestra la imagen control de cerraduras con las respectivas coordenadas de cada botón, se deben controlar 3 columnas y 3 fila.

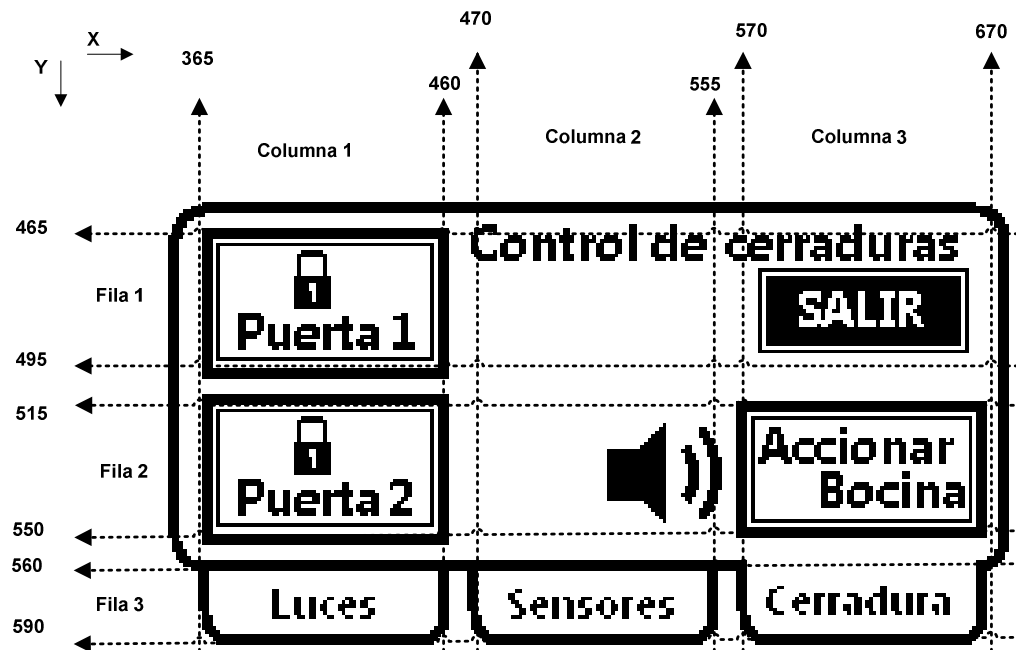


Figura 157: Coordenadas de la imagen Control de cerraduras

El siguiente fragmento de código, muestra el sub proceso que identifica las columnas, filas y los botones o también denominados áreas de presión de la imagen Control de cerraduras.

```
sub procedure Touch_cerraduras()
    x=xpos 'x
    y=ypos 'y
    'Columnas
    if (x >= 365) and (x < 460) then columna=1 end if
    if (x >= 470) and (x < 555) then columna=2 end if
    if (x >= 570) and (x < 670) then columna=3 end if
    if xpos = 0 then columna=0 end if
    'Filas
    if (y >= 465) and (y < 495) then fila=1 end if
    if (y >= 515) and (y < 550) then fila=2 end if
```

```

if (y >= 560) and (y < 590) then fila=3 end if
if ypos = 0 then fila=0 end if
'botones (areas de presion)
if (columna=1) and (fila=1) then dato=1 end if
if (columna=3) and (fila=1) then dato=2 end if
if (columna=1) and (fila=2) then dato=3 end if
if (columna=3) and (fila=2) then dato=4 end if
if (columna=1) and (fila=3) then dato=5 end if
if (columna=2) and (fila=3) then dato=6 end if
if (columna=3) and (fila=3) then dato=7 end if
if (columna=0) and (fila=0) then dato=16 end if
end sub

```

Una vez creado el código de cada panel solo se debe comparar el dato que reconoce cada botón de cada uno de ellos, para realizar la acción correspondiente, que sería enviar datos al servidor o recibir datos desde el servidor, para esto se usan dos sub procedimientos envío y recepción de datos con la ayuda del método Usart\_Write\_Text, en el siguiente fragmento de código, se muestran ambos sub procedimientos.

```

sub procedure envio_datos()
    Usart_Write_Text(envio)
    enter
    delay_ms(300)
end sub

sub procedure recibo_datos()
    while salir3 = 0
        if Usart_Data_Ready = 1 then
            recibo = Usart_Read
            'Usart_Write(recibo)
            'enter
            if recibo = 112 then
                salir3=1
            end if
            if recibo = 111 then
                salir3=1
            end if
        end if
    end while
end sub

```

Un problema que se presenta es la de separar los datos enviados al servidor por medio de un ENTER para esto se usa es siguiente procedimiento.

```
sub procedure enter()  
Usart_Write(0x0D)  
end sub
```

La validación del usuario se realiza enviando el código y tipo de usuario ingresado al servidor y este responde con una 's' si es correcto o con una 'n' si es incorrecto, se comparan estos caracteres y el usuario del dispositivo inalámbrico tiene o no acceso a los demás paneles.

Al momento de compilar se observa que la memoria ROM del microcontrolador esta usada casi en su totalidad por lo que colocar otra imagen mas es imposible.

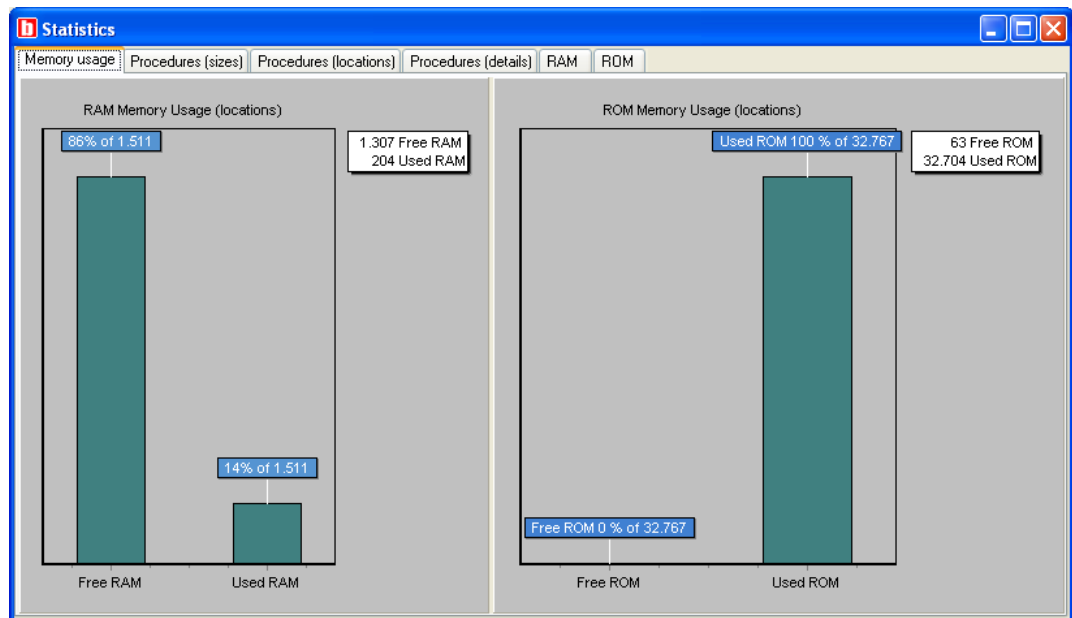


Figura 158: Estadísticas de la memoria RAM Y ROM usadas por el microcontrolador

Para mayor información sobre la programación del microcontrolador ubicado en el módulo GLCD ver el código en el anexo digital CD-ROM:\CIRCUITOS\ GLCD 240x128\ GLCDXbee.pbp

298



Figura 160: Fotografía Pantalla Iniciando



Figura 161: Fotografía Pantalla ingresar tipo de usuario



Figura 162: Fotografía Pantalla ingresar código del usuario



Figura 163: Fotografía Pantalla control de iluminación.



Figura 164: Fotografía monitoreo de sensores.



Figura 165: Fotografía Pantalla Control de cerraduras



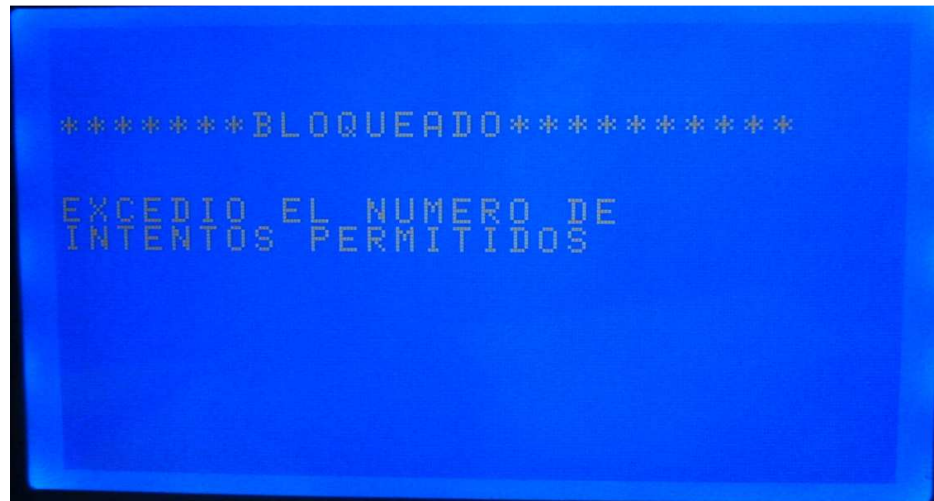


Figura 166: Fotografía Pantalla Bloqueo excedió el número de intentos permitidos

### 5.3.2. APLICACIÓN MÓVIL BLUETOOTH J2ME

El desarrollo de la aplicación Bluetooth se realiza utilizando la herramienta Mobile Processing 0007, ver **figura #** .

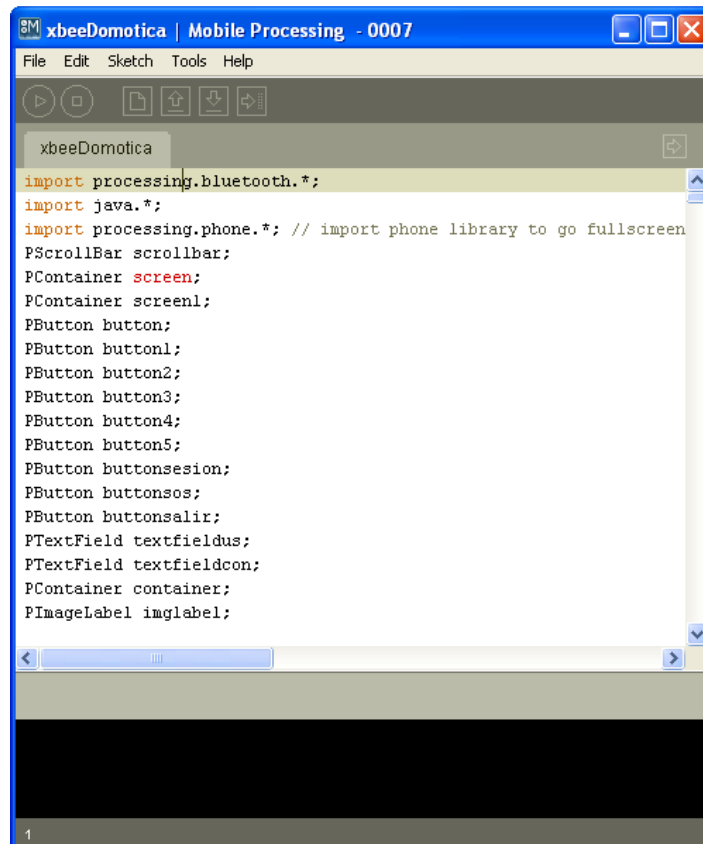


Figura 167: Desarrollador Mobile Processing 007

Se usan las siguientes librerías:

`import processing.bluetooth.*;` Librería para el descubrimientos de dispositivos cercanos y creación de servicios con Bluetooth.

`import processing.phone.*;` Librería para el uso de eventos del teléfono móvil por ejemplo la vibración.

Para compilar y ejecutar un programa realizado en Mobile Processing es necesario el desarrollador Wireless Toolkit y se debe configurar la ruta en las preferencias de Mobile Processing ej.: C:\WTK2.5.2\_01.

La creación de botones, TextField o label se realiza directamente por código, se pueden colocar fondos .png para diferenciar las distintas pantallas, en las siguientes figuras, se muestran los 3 fondos que se van a usar en la aplicación, el fondo inicial, fondo de registro de usuarios y el fondo de control domótico.



Figura 168: Fondo bg.png



Figura 169: Fondo us.png



Figura 170: Fondoilu.png

Al iniciar la aplicación el sistema da la bienvenida ver Figura 171.

Bluetooth.EVENT\_DISCOVER\_DEVICE : Busca dispositivos bluetooth.

Bluetooth.EVENT\_DISCOVER\_SERVICE : Descubre los servicios que están ofrecidos en este caso se busca el del Puerto COM serial.

Bluetooth.EVENT\_CLIENT\_CONNECTED: conecta al cliente con el servidor.

```
if (library == bt) {
  switch (event) {
    case Bluetooth.EVENT_DISCOVER_DEVICE:
      msg = "Dispositivo encontrado : " + ((Device) data).address + "...";
      break;
    case Bluetooth.EVENT_DISCOVER_DEVICE_COMPLETED:
      msg = "Encontrado " + length((Device[]) data) + " dispositivos, buscando puertos
seriales...";
      break;
    case Bluetooth.EVENT_DISCOVER_SERVICE:
      msg = "Encontrado puerto serial : " + ((Service[]) data)[0].device.address + "...";
      break;
    case Bluetooth.EVENT_DISCOVER_SERVICE_COMPLETED:
      services = (Service[]) data;
      msg = "Busqueda completa seleccione un puerto ";
      break;
    case Bluetooth.EVENT_CLIENT_CONNECTED:
      c = (Client) data;
      msg = "Cliente conectado";
```

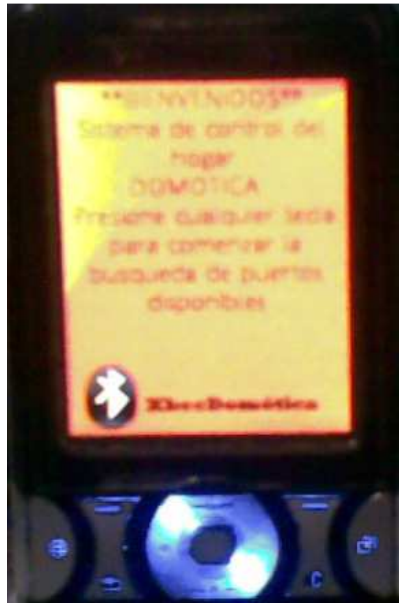


Figura 171: Fotografía Pantalla Bienvenida

En el siguiente fragmento de código, se muestra la creación de la primera pantalla que permite el inicio de sesión a un usuario, consta de 2 cajas de texto y 3 botones, se carga el fondo de pantalla us.png ver Figura 172.

```
void botones_sesion() {

    PImage img = loadImage("us.png");
    imglabel = new PImageLabel(img);
    imglabel.setBounds(0,0,img.width,img.height);
    screen.add(imglabel);

    textfieldus = new PTextField("");
    textfieldus.calculateBounds((width / 2) - 80,37, 100,
    PContainer.HEIGHT_UNBOUNDED);
    screen.add(textfieldus);

    textfieldcon = new PTextField("");
    textfieldcon.calculateBounds((width / 2) - 80,80,100,
    PContainer.HEIGHT_UNBOUNDED);
    textfieldcon.password = true;
    screen.add(textfieldcon);
}
```

```

buttonsesion = new PButton("Iniciar Sesion" );
buttonsesion.calculateBounds((width / 2) - 80, 110, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(buttonsesion);

buttonsos = new PButton("EMERGENCIA" );
buttonsos.calculateBounds((width / 2) - 80, 160, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(buttonsos);

buttonsalir = new PButton("SALIR" );
buttonsalir.calculateBounds((width / 2) - 80, 190, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(buttonsalir);
}

```



Figura 172: Fotografía Pantalla Inicio de sesión

Cuando se hace clic en el botón iniciar sesión la aplicación envía el nombre de usuario y contraseña separados por un “ . ” al servidor y espera que le devuelva ‘s’ si el usuario es correcto o ‘n’ si el usuario es incorrecto.

```

if (library == buttonsesion) {
    comando="\n";
    byte m[] = comando.getBytes();
    c.write(m);
    c.flush();
    b4=false;
    llega=0;
}

```

Envía el usuario y la contraseña al servidor separada por un punto.

```
usuario = textfieldus.text + " . ";
contra = textfieldcon.text + "\n";
byte us[] = usuario.getBytes();
c.write(us);
byte co[] = contra.getBytes();
c.write(co);
c.flush();
```

Escucha el canal de comunicaciones en busca de la respuesta a la petición realizada si el usuario es correcto o no.

```
while(llego==0){
    if (c.available() > 0) {
        nu=c.read ();
        llego= 1;
    }
}
```

Compara el dato enviado por el servidor 's' en decimal 115 o 'n' en decimal 110.

```
if(nu==115 ){
    inicio=1;
    sesion=1;
    etiqueta=false;
    estados=true;
    setup();
}
if(nu==110 ){
    //contraseña incorrecta
    myPhone.vibrate (600);
    b4=true;
}
}
```

Para el botón Emergencia se programa el envío de una alerta al servidor en este caso al presionarlo envía la palabra “EMERGENCIA” y produce una vibración al celular.

```
if (library == buttonsos) {  
    comando="EMERGENCIA"+ "\n";  
    byte sos[] = comando.getBytes();  
    c.write(sos);  
    c.flush();  
    myPhone.vibrate (1000);  
}
```

Para el botón salir solo llamo al método exit () y listo

```
if (library == buttonsalir) {  
    exit ();  
}
```

En el siguiente fragmento de código, se muestra la creación de la pantalla de control se usa el fondo de pantalla ilu.png y consta de 6 botones y 4 label que muestran el estado de los sensores monitoreados ver Figura 173.

```
void botones_menu() {  
    PImage img = loadImage("ilu.png");  
    imglabel = new PImageLabel(img);  
    imglabel.setBounds(0,0,img.width,img.height);  
    screen.add(imglabel);  
    button = new PButton("Sala " );  
    button.calculateBounds(0, 30, width - 8,  
        PContainer.HEIGHT_UNBOUNDED);  
    screen.add(button);  
  
    button1 = new PButton("Exterior " );  
    button1.calculateBounds(75, 30, width - 8,  
        PContainer.HEIGHT_UNBOUNDED);  
    screen.add(button1);  
}
```

```

button2 = new PButton("Cocina " );
button2.calculateBounds(0, 60, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(button2);

button3 = new PButton("Dormitorio " );
button3.calculateBounds(75, 60, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(button3);

button4 = new PButton("Act. Sensores" );
button4.calculateBounds(10, 90, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(button4);
button5= new PButton("Salir" );
button5.calculateBounds(10, 180, width - 8,
PContainer.HEIGHT_UNBOUNDED);
screen.add(button5);
}

```



Figura 173: Fotografía Pantalla control iluminación y monitoreo.

La programación de los 4 botones de control en este caso “SALA”, “EXTERIOR”, “COCINA” y “DORMITORIO” se resume en el envío de un comando para que el servidor lo interprete y realice una acción determinada.



## SALA

```
if (library == button) {  
    /// if the button sent a library event, it was pressed!  
    comando="A"+"\\n";  
    byte cmd[] = comando.getBytes();  
    c.write(cmd);  
    c.flush();  
    myPhone.vibrate (100);  
}
```

## EXTERIOR

```
if (library == button1) {  
    /// if the button sent a library event, it was pressed!  
    comando="B"+ "\\n";  
    byte cmd2[] = comando.getBytes();  
    c.write(cmd2);  
    c.flush();  
    myPhone.vibrate (100);  
}
```

## COCINA

```
if (library == button2) {  
    /// if the button sent a library event, it was pressed!  
    comando="C"+ "\\n";  
    byte cmd3[] = comando.getBytes();  
    c.write(cmd3);  
    c.flush();  
    myPhone.vibrate (100);  
}
```

## DORMITORIO

```
if (library == button3) {  
    comando="D"+ "\\n";  
    byte cmd4[] = comando.getBytes();  
    c.write(cmd4);  
    c.flush();  
    myPhone.vibrate (100);  
}
```

El botón “Estados” envía una petición al servidor cada vez que se hace clic sobre él, y le devuelve el estado del sensor solicitado.

```
if (library == button4) {  
    todos=0;  
    comando="ESTADO"+ "\n";  
    byte cmd5[] = comando.getBytes();  
    c.write(cmd5);  
    c.flush();  
    myPhone.vibrate (100);  
    while(todos==0){  
        if (c.available() > 0) {  
            nu=c.read ();  
            cont=cont+1;  
            actuadores();  
            if(cont>3){  
                cont=0;  
                todos= 1;  
            }  
        }  
    }  
}
```

### **5.3.3. SERVIDOR PRINCIPAL**

El servidor principal como se dijo en el capítulo 3, diseño de software es desarrollado con netbeans IDE 6.5.1 usando Java web JSF.

#### **5.3.3.1. Estándares de programación**

Un estándar de programación es una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto cualquiera de las personas involucradas en el mismo tenga acceso y comprendan el código.

En otras palabras define la escritura y organización del código fuente de un programa. Además el seguir un estándar de programación te facilita como programador la modificación de tu propio código fuente aunque no estés trabajando en un equipo.

#### 5.3.3.1.1. Estándar de variables

Los nombres de las variables que se manejan en el servidor son de fácil entendimiento porque se usan nombres claves para definir el estado de los sensores o actuadores del sistema, estos nombres claves son los mismos que se usan para definirlos por ejemplo:

Focosala  
Cerradura1  
Mov1

#### 5.3.3.1.2. Estándar de componentes de la interfaz

Los componentes de la interfaz, se refiere a los componentes utilizados en la interfaz gráfica de la aplicación por ejemplo las cajas de texto, botones, label, etc.

Componente	Nomenclatura	Descripción
Button	btn_(nombre)	Representa a los botones
Label	lbl__(nombre)	Representa a las etiquetas
Text Field	txt_(nombre)	Representa a las cajas de texto
Checkbox	chk_(nombre)	Representa a los Checkbox
Calendar	cal_(nombre)	Representa al calendario
Listbox	lst_(nombre)	Representa a las listas
Drop Down List	cb_(nombre)	Representa a los Combobox
Grid Panel	gp__(nombre)	Representa a los paneles
Table	tbl_(nombre)	Representa a las tablas
Tree_Node	tree_(nombre)	Representa al Tree View

Tabla 42 : Estándar de componentes de la interfaz

Usando un correcto estándar de componentes de la interfaz de forma ordenada se facilita la programación, modificación, etc.

### 5.3.3.2. Módulos

Con el fin de seguir el mismo orden del diseño de software del servidor principal se ha dividido en 6 módulos de programación.

#### 5.3.3.2.1. Módulo Servidor SMS

Se crea una clase `smsx.java` para que escuche al puerto COM en donde está conectado el modem GSM en este caso el celular Nokia 3220b y también es necesario la creación de un hilo que trabaje independientemente enviando el comando AT de lectura SMS para comprobar si llegan mensajes nuevos.

```
import javax.comm.*;
```

```
public class smsx extends Thread implements Runnable,  
SerialPortEventListener {
```

Se instancia a la clase `XbeeTouch`

```
clases.XbeeTouch xbt = new clases.XbeeTouch();
```

Para establecer la comunicación entre la clase `smsx` y el modem GSM se usa el siguiente fragmento de código.

Para abrir el puerto de comunicación se envía un comando en blanco el método “`enviar`” recibe un parámetro de tipo `STRING` que lo envía por el puerto COM hacia el modem GSM.

Se define el número de puerto 11 para establecer la comunicación . 2400 de velocidad, bits de datos 8, bits de parada 1 y control de flujo ninguno.

```

String line1 ;
line1 = comando + "\r\n";
portList = CommPortIdentifier.getPortIdentifiers();
while (portList.hasMoreElements()) {
    portId = (CommPortIdentifier) portList.nextElement();
    if (portId.getPortType() == CommPortIdentifier.
        PORT_SERIAL) {
        System.out.println("Enviando comando.....\r\n");
        if ( portId.getName().equals("COM11")) {
            salida = serialPort.getOutputStream();
            serialPort.setSerialPortParams(
                2400,
                SerialPort.DATABITS_8,
                SerialPort.STOPBITS_1,
                SerialPort.PARITY_NONE);
            salida.write(line1.getBytes());
            salida.flush();
        }
    }
}

```

Antes de enviar cualquier comando al modem GSM es necesario inicializarlo con este comando.

```

enviar("AT+CMGF=1");

```

Para leer un mensaje de texto se tiene que enviar el comando:

```

enviar("AT+CMGL=\"REC UNREAD\"");

```

Esto se debe realizar de manera periódica por lo que es necesario crear un hilo que envíe este comando cada t segundos, este hilo debe trabajar de manera paralela al sistema.

```

public class HiloLeeSms extends Thread{
    clases.smsx sms = new clases.smsx();
    int bandera ;
    @Override
    public void run()

```

```

{
    bandera = 0 ;
    while(true){
        try {
            Thread.sleep(2000);
            bandera = sms.pausar();
            if(bandera==0){
                sms.enviar("AT+CMGL=\"REC UNREAD\"");
            }
        } catch (Exception e) {
            log("Exception Message!!!", e);
        }
    }
}
}

```

Para escuchar el Puerto COM determinado se usa el siguiente código, cuando llegue un dato al puerto este lo almacena en la variable vector líneas, luego se compara esta variable, de acuerdo a la posición en la que llega el dato y se realiza la acción específica.

```

public void serialEvent(SerialPortEvent event)
{
    switch (event.getEventType())
    {
        case SerialPortEvent.BI:
        case SerialPortEvent.OE:
        case SerialPortEvent.FE:
        case SerialPortEvent.PE:
        case SerialPortEvent.CD:
        case SerialPortEvent.CTS:
        case SerialPortEvent.DSR:
        case SerialPortEvent.RI:
        case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
            break;
        case SerialPortEvent.DATA_AVAILABLE:
            {
                BufferedReader reader = new BufferedReader(new
                InputStreamReader(entrada));
            }
    }
}

```

```

String line = "";
try
{
    c = 0 ;
    while ( (line = reader.readLine()) != null)
    {
        lineas.add(line);
    }
}

```

Para enviar un SMS se define el número de celular al que va a ser enviado y el contenido del mensaje.

Se tienen que enviar de forma serial estas tres líneas que contienen la estructura del comando AT para envío de SMS.

```

String line1 = "AT+CMGS="+ número + "\r\n";
String line2 = msj + "\r\n";
String line3 = "\u001a\r\n";

```

Una vez que el servidor SMS esté pendiente de mensajes nuevos cada t segundos es cuestión de comparar los mensajes nuevos y realizar acciones específicas de control de dispositivos finales.

Por ejemplo si recibe el comando “prende sala” el sistema verifica el estado actual del foco de la sala si está apagado envía el comando al Xbee coordinador para que encienda el foco de la sala.

```

if (pos4.equalsIgnoreCase("prende sala")){
    if(xbt.foco_sala () == 0){
        xbt.medio_sms(cod_usuariosms);
        xbt.enviadato("A");
    }
}

```

Una seguridad implementada en el servidor SMS es que cuando llega un mensaje nuevo se comprueba que el número de remitente exista en la base de datos, si existe entonces se realiza la acción caso contrario no hace nada. Se realiza la comparación de estos datos por medio de la base de datos.

```
reg = cxn.Consulta("SELECT cod_usuario FROM usuarios where  
CELULAR='"+celularc.trim()+"");
```

Para la gestión de cerraduras el usuario debe ser administrador y enviar su código de usuario junto con el comando de apertura de cerradura, una vez obtenidos estos datos se compara con los de la base de datos.

```
reg = cxn.Consulta("SELECT cod_usuario FROM usuarios where  
cod_usuario='"+codigo_usuario+"' and CELULAR='"+celularc.trim()+"");
```

Para mayor información sobre la programación del módulo servidor SMS ver el código en el anexo digital CD-ROM:\ServidorXbeedomótica

#### **5.3.3.2.2. Módulo Servidor Bluetooth**

Una vez que las configuraciones en el dispositivo bluetooth cliente están realizadas y está emparejado con el dispositivo bluetooth USB.

Se crea una clase Bluetooth.java para que establezca comunicación con el puerto COM en donde está conectado el dispositivo Bluetooth USB para este propósito se usa la librería javax.comm.

```
import javax.comm.*;
```

```
public class Bluetooth extends Thread implements Runnable,  
SerialPortEventListener {
```



Se instancia a la clase XbeeTouch para tener acceso a sus métodos y enviar datos al Xbee coordinador.

```
clases.XbeeTouch xbt = new clases.XbeeTouch();
```

Para establecer comunicación entre la clase Bluetooth y el dispositivo Bluetooth USB se usa el siguiente fragmento de código.

Para abrir el puerto de comunicación se envía un comando en blanco el método “enviadata” recibe un parámetro de tipo STRING que lo envía por el puerto COM hacia el dispositivo Bluetooth USB.

Se define el número de puerto 17 para establecer la comunicación. 9600 de velocidad, bits de datos 8, bits de parada 1 y control de flujo ninguno.

```
public void enviadata (String envio){
    String line1 = envio;
    portList = CommPortIdentifier.getPortIdentifiers();
    while (portList.hasMoreElements()) {
        portId = (CommPortIdentifier) portList.nextElement();
        if (portId.getPortType() == CommPortIdentifier.
            PORT_SERIAL) {
            System.out.println("Enviando comando BT.....\r\n");
            if ( portId.getName().equals("COM17")) {
                salida = serialPort.getOutputStream();
                serialPort.setSerialPortParams(
                    9600,
                    SerialPort.DATABITS_8,
                    SerialPort.STOPBITS_1,
                    SerialPort.PARITY_NONE);
                salida.write(line1.getBytes());
                salida.flush();
            }
        }
    }
}
```

Este método recibe un STRING el cual es el mensaje que se quiere enviar por medio del puerto COM 17 al dispositivo Bluetooth USB y este a su cliente.

Para que la clase Bluetooth detecte datos entrantes en su búfer se escriben estas líneas de código.

```
public void serialEvent(SerialPortEvent event)
{
    switch (event.getEventType())
    {
        case SerialPortEvent.BI:
        case SerialPortEvent.OE:
        case SerialPortEvent.FE:
        case SerialPortEvent.PE:
        case SerialPortEvent.CD:
        case SerialPortEvent.CTS:
        case SerialPortEvent.DSR:
        case SerialPortEvent.RI:
        case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
            break;
        case SerialPortEvent.DATA_AVAILABLE:
            {
                BufferedReader reader = new BufferedReader(new
                InputStreamReader(entrada));
                String line = "";
                try
                {
```

Una vez que detecta datos entrantes los lee línea tras línea hasta que no haya más, los almacena en una variable “line” que los va comparando según se crea conveniente ya sea para activar o desactivar un actuador o para validar un usuario.

```
        while ( (line = reader.readLine()) != null)
        {
```

```

estado = line.trim();
int n = estado.indexOf('.');
System.out.println("indice es : " + n);
if (n>0){
    System.out.println("Verificamos usuario y contraseña");
    verifica_clave(n);
}
if(estado.equalsIgnoreCase("A")|| estado.equalsIgnoreCase("B") ||
estado.equalsIgnoreCase("C")||estado.equalsIgnoreCase("D")||estado.e
qualsIgnoreCase("EMERGENCIA")){
    System.out.println("Verificamos el dato");
    verifica_dato();
}
if (estado.equalsIgnoreCase("ESTADO")){
    System.out.println("Verificamos el estado actuadores");
    estado_actuadores ();
}
}

```

Para comparar la contraseña y el nombre de usuario con la base de datos se usa este fragmento de código, realiza la consulta con la contraseña y el nombre de usuario si devuelve un valor, el usuario está correcto caso contrario el usuario y/o contraseña son inválidos.

El nombre de usuario y la contraseña vienen juntos en un mensaje separados por un espacio y un punto, es por esta razón que se debe separar con ayuda de un substring para tenerlos de forma individual y realizar las respectivas comparaciones

```

public void verifica_clave(int indice){
    b = 0 ;
    String usuario = estado.substring(0, indice - 1);
    String contra = estado.substring(indice + 1 ,estado.length() );
    try
    {

```

```

reg = cxn.Consulta("SELECT cod_usuario FROM usuarios where
contra='"+contra.trim()+"' and nombre='"+usuario.trim()+"'");
while(reg.next())    {
    System.out.println("imprime");
    String codigo_us= reg.getString("cod_usuario");
    System.out.println("cod usuario = " + codigo_us );
        tiempo(2000);
        enviadato("s");
    }

```

Una vez que el servidor Bluetooth esté pendiente de los datos nuevos que lleguen a su búfer, es cuestión de comparar estos datos y realizar acciones específicas de control de dispositivos finales.

Por ejemplo si recibe el dato “A” el sistema envía este dato al coordinador para que este lo difunda entre sus clientes y se realice la acción respectiva en este caso es apagar o encender el foco de la sala.

```

if (estado.equalsIgnoreCase("A") ){
    xbt.medio_bluetooth(codusuario_bt);
    xbt.enviadato("A");
}

```

La Aplicación Bluetooth también pide información acerca del estado de los diferentes sensores, realiza un monitoreo de la siguiente manera.

Primero consulta el estado actual del sensor por medio de la clase instanciada xbt y devuelve enviando un código según sea su estado y el tipo de sensor.

```

public void estado_sensores (){
    int mov1 = xbt.detector_mov1();
    int mov2 = xbt.detector_mov2();
    int sm1 = xbt.sensor_magnetico1();
    int sm2 = xbt.sensor_magnetico1();

```

```

if( mov1 == 1){
    enviadato("1");
    tiempo(500);
}
if( mov1 == 0){
    enviadato("2");
    tiempo(500);
}
if( mov2 == 1){
    enviadato("3");
    tiempo(500);
}
if( mov2 == 0){
    enviadato("4");
    tiempo(500);
}

if( sm1 == 0){
    enviadato("6");
    tiempo(500);
}
if( sm1 == 1){
    enviadato("5");
    tiempo(500);
}
if( sm2 == 0){
    enviadato("8");
    tiempo(500);
}
if( sm2 == 1){
    enviadato("7");
    tiempo(500);
}
}
}

```

Para mayor información sobre la programación del módulo servidor Bluetooth ver el código en el anexo digital CD-ROM:\ServidorXbeedomótica

#### **5.3.3.2.3. Módulo Servidor Zigbee**

Este módulo es el centro del servidor principal es el que controla al coordinador Zigbee y por lo tanto a todos los módulos de comunicaciones del sistema que contienen todos los dispositivos finales, incluyendo el módulo GLCD pantalla táctil.

Gestiona la comunicación entre los módulos de comunicaciones y el coordinador, controla el módulo GLCD su inicio de sesión validando los datos de usuario ingresados por este medio, monitorea y controla todos los dispositivos finales.

Se crea una clase XbeeTouch.java para que establezca comunicación con el puerto COM en donde está conectado la interface serial Xbee USB para este propósito se usa la librería javax.comm.

```
import javax.comm.*;
```

```
public class XbeeTouch extends Thread implements Runnable,  
SerialPortEventListener {
```

Se instancia a la clase “Correo” para tener acceso a sus métodos y enviar notificaciones de correo electrónico a los usuarios que se requiera.

```
clases.EnviaCorreo correo = new EnviaCorreo();
```

Para establecer comunicación entre la clase XbeeTouch y la interface serial Xbee USB se usa el siguiente fragmento de código.

Para abrir el puerto de comunicación se envía un comando en blanco, el método “enviadata” recibe un parámetro de tipo STRING que lo envía por el puerto COM hacia la interface serial Xbee USB.

Se define el número de puerto 14 para establecer la comunicación. 9600 de velocidad, bits de datos 8, bits de parada 1 y control de flujo ninguno.

```
public void enviadata (String envio){

    String line1 = envio;
    portList = CommPortIdentifier.getPortIdentifiers();
    while (portList.hasMoreElements()) {
        portId = (CommPortIdentifier) portList.nextElement();
        if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {
            System.out.println("Enviando comando.....\r\n");
            if ( portId.getName().equals("COM14")) {
                try {
                    serialPort = (SerialPort) portId.open("SimpleWriteApp", 2000);
                    try {
                        serial = serialPort;
                        entrada = serial.getInputStream();
                        serial.addEventListener(this);
                        serial.notifyOnDataAvailable(true);
                    } catch (PortInUseException e) {
                        System.out.println("Puerto en uso " + e);
                    }
                }
                salida = serialPort.getOutputStream();
                serialPort.setSerialPortParams(
                    9600,
                    SerialPort.DATABITS_8,
                    SerialPort.STOPBITS_1,
                    SerialPort.PARITY_NONE);
                salida.write(line1.getBytes());
                salida.flush();
            }
        }
    }
}
```

Este método recibe un STRING el cual, es el mensaje que se quiere enviar por medio del puerto COM 14 a la interface serial Xbee USB y esta a sus nodos clientes.

Para que la clase “XbeeTouch” detecte datos entrantes en su búfer se escriben estas líneas de código.

```
public void serialEvent(SerialPortEvent event)
{
    switch (event.getEventType())
    {
        case SerialPortEvent.BI:
        case SerialPortEvent.OE:
        case SerialPortEvent.FE:
        case SerialPortEvent.PE:
        case SerialPortEvent.CD:
        case SerialPortEvent.CTS:
        case SerialPortEvent.DSR:
        case SerialPortEvent.RI:
        case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
            break;
        case SerialPortEvent.DATA_AVAILABLE:
            {
                BufferedReader reader = new BufferedReader(new
                InputStreamReader(entrada));
                String line = "";
                try
                {
```

Una vez que detecta datos entrantes los lee línea tras línea hasta que no hayan más, los almacena en una variable “line” que los va comparando según se crea conveniente ya sea para activar o desactivar un actuador o para validar un usuario a petición del Módulo GLCD.



```

while ( (line = reader.readLine()) != null)
{
    estado = line.trim();
    System.out.println("Este es el estado "+ estado);
    if (estado.equalsIgnoreCase("n") ||
estado.equalsIgnoreCase("s") || estado.equalsIgnoreCase("o") ||
estado.equalsIgnoreCase("p") ){
        llego=0;
        System.out.println("llego a cero");
        estado ="nada";
    }
    else {
        llego=1;
        System.out.println("llego a uno");
    }
    if (estado.equalsIgnoreCase("estado1") ||
estado.equalsIgnoreCase("puerta1")){
        peticion_estado= 1;
    }
    verifica_dato();
}
serialPort.close();
break;

```

Para comparar el código y tipo de usuario a petición del módulo GLCD con la base de datos se usa este fragmento de código, realiza la consulta con el código y tipo del usuario si devuelve un valor, el usuario está correcto caso contrario el código y/o tipo son inválidos.

```

reg = cxn.Consulta("SELECT cod_usuario FROM usuarios where
cod_usuario='"+contra+"' and cod_tipo='"+tipou+"'");
while(reg.next())
{
    String contrasena= reg.getString("cod_usuario");

    codusuario_txb=contrasena;
    tiempo(1000);
}

```

```

        this.enviadato("s");
        llego=0;
        b = 1 ;
        bad=0;
    }
    if ( b == 0){
        System.out.println("no existe el usuario");
        tiempo(1000);
        this.enviadato("n");
        llego=0;
        //bad=0;
    }
} catch(SQLException ex){System.out.println(ex);}
}

```

Si el usuario existe envió la letra “s” por medio del puerto serial hasta el coordinador y este al módulo GLCD, caso contrario envió la letra “n” quiere decir usuario incorrecto.

Nota.- El código de usuario no es la contraseña, en la tabla usuarios de la base de datos hay dos campos el código y el campo contraseña, por motivos de factibilidad en el módulo GLCD, se usa el código del usuario como clave de ingreso porque en la pantalla Touch solo se permite la digitación de números no de caracteres, también estos números deben identificar al usuario es decir, ser únicos por lo tanto se usa el código del usuario que es numérico y además es clave principal.

El módulo GLCD también realiza peticiones de monitoreo al servidor y este le envía el estado de los diferentes sensores o actuadores, le envía una ‘p’ si están activados o una ‘o’ si están desactivados.

```

if (estado.equalsIgnoreCase("SALA")){
    if (foco_sala==1){
        tiempo(1000);
        this.enviadato("p");
    }
    if (foco_sala==0){
        tiempo(1000);
        this.enviadato("o");
    }
}
if (estado.equalsIgnoreCase("EXT")){
    if (foco_exterior==1){
        tiempo(1000);
        this.enviadato("p");
    }
    if (foco_exterior==0){
        tiempo(1000);
        this.enviadato("o");
    }
}
}

```

De la misma manera realiza el envío del estado actual de todos los sensores y actuadores del sistema.

Al iniciar el servidor se desconoce el estado de los sensores o actuadores de cada uno de los módulos, por lo que se realiza una petición a cada uno de ellos con el fin de conocer su estado actual.

```

public void xbee1(){
    inicializacion=false;
    this.enviadato("x");
    System.out.println("Seteando Xbee1 ");
}
public void xbee2 (){
    inicializacion=false;
    this.enviadato("y");
    System.out.println("Seteando Xbee2 ");
}
public void xbee3 (){

```

```

        inicializacion=false;
        this.enviadato("Z");
        System.out.println("Seteando Xbee3 " );
    }
    public void xbeefinal (){
        inicializacion=true;
        System.out.println("Seteo de Xbee finalizado " );
    }

```

Para controlar los actuadores desde el módulo GLCD se reciben datos como “SALA”, “ETX”, etc.

Se interpreta y se envía el comando correcto al coordinador y este a sus clientes.

El código de medio que aquí se asigna es el número tres que corresponde a Control Touch.

```

if (estado.equalsIgnoreCase("SALA")){
    this.enviadato("A");
    codmedio = "3";
}
if (estado.equalsIgnoreCase("EXT")){
    this.enviadato("B");
    codmedio = "3";
}
if (estado.equalsIgnoreCase("COCI")){
    this.enviadato("C");
    codmedio = "3";
}

```

En esta clase se guardan los eventos ocurridos en el historial de la base de datos para esto en necesario instanciar la clase “consultasbd” que permite insertar nuevas filas en la base de datos.

```

consultasbd cbdd =new consultasbd();
public void insertar_histo(){
    obtener_fecha();

```

```

codpunto=estado.substring(0,3);
if(inicializacion){
cbdd.historial(codusuario, codpunto, codmedio, estvariable, fecha,
hora);
}
codmedio="5";
}

```

La hora y la fecha se obtienen usando este fragmento de código.

```

public void obtener_fecha(){
    Date fechat=new Date();
    Format formatter;
    formatter = new SimpleDateFormat("yyyy-MM-dd");
    fecha = formatter.format(fechat);
    hora=fechat.toString().substring(11,19);
}

```

Las variables dentro del método historial son previamente inicializadas en el evento ocurrido, por ejemplo si se apagó el foco de la sala llega la notificación a este módulo servidor Zigbee en forma de código de comunicación “1120”.

Aquí se identifica el usuario que ocasiono este evento, el medio por el cual se realizo Web, Bluetooth, Control Touch, Físico o SMS, El punto afectado es decir, el dispositivo final y el estado en que se encuentra .

```

if (estado.equalsIgnoreCase("1120")){
    foco_sala=0;
    bandera2=true;
    System.out.println("Foco Sala apagado");
    tipo_usu();
    estvariable="APAGADO";
    if(inicializacion){
    if(iluminacion){insertar_histo();}
    consulta_usuario();
}

```

Dentro de este evento se envía un correo de notificación a los usuarios autorizados a recibir dicha notificación con la siguiente información.

```
if(todoMail && iluminacionMail){enviar_mail(" Notificación Iluminación
","El foco de la sala ha sido apagado \n Fecha : "+fecha+"\n Hora :
"+hora+"\n Por medio del servicio "+medio + "\n Por el usuario : " +
nomb_usuario);}
```

```
public void enviar_mail(String asunto , String mensaje){
EnviarCorreo.sendMail("zigbeedomotica@hotmail.com","tesisups","smt
p.gmail.com","465","true","true",true,"javax.net.ssl.SSLSocketFactory","f
alse",mail1,mail2,mail3,asunto,mensaje);
}
```

Dentro de esta clase también se encuentran los modos Ahorros de energía y modo alarma sonora.

Para el modo ahorro de energía se usa el siguiente código, solo se puede gestionar este modo por medio de sensores de movimiento y sensor fotovoltaico, alteran el estado de tres focos cocina, sala y exterior.

```
public void MaeSensores(){
if(MaeMov){
System.out.println("ENTRAMOS A MODO AHORRO DD
ENERGIAA");
if(detector_mov1() == 0 && foco_sala == 1 && bandera1){
bandera1 = false;
bandera2=true;
System.out.println("APAGA FOCO SALA");
tiempo(600);
enviadata("A");
}
if(detector_mov1() == 1 && foco_sala == 0 && bandera2){
```

```

bandera1=true;
bandera2=false;
System.out.println("ENCIENDE FOCO SALA");
tiempo(600);
enviadoato("A");
}
if(detector_mov2() == 0 && foco_cocina== 1 && bandera3){
bandera3 = false;
bandera4=true;
System.out.println("APAGA FOCO COCINA");
tiempo(600);
enviadoato("C");
}
if(detector_mov2() == 1 && foco_cocina == 0 && bandera4){
bandera3=true;
bandera4=false;
System.out.println("ENCIENDE FOCO COCINA");
enviadoato("C");
}
}
}
public void Maefoto(){
    if(MaeFoto ){
        if(detector_luz() == 0 && foco_exterior == 1 && bandera5){
            bandera5 = false;
            bandera6=true;
            System.out.println("APAGA FOCO EXTERIOR");
            tiempo(600);
            enviadoato("B");
        }
        if(detector_luz() == 1 && foco_exterior == 0 && bandera6){
            bandera5=true;
            bandera6=false;
            System.out.println("ENCIENDE FOCO EXTERIOR");
            tiempo(600);
            enviadoato("B");
        }
    }
}
}

```

Para el modo alarma sonora se enciende la bocina si los detectores de movimiento, humo, gas o magnéticos son activados.

```
public void ModoA(){
    if(modoA==1){
        if(MaMov){
            if(detector_mov1() == 1 && bocina == 0 && band_b1){
                band_b1 = false;
                System.out.println("ENCIENDE BOCINA");
                tiempo(600);
                enviadato("H");
            }
            if(detector_mov1() == 0){
                band_b1=true;
            }
            if(detector_mov2() == 1 && bocina == 0 && band_b2){
                band_b2 = false;
                System.out.println("ENCIENDE BOCINA");
                tiempo(600);
                enviadato("H");
            }
            if(detector_mov2() == 0){
                band_b2=true;
            }
        }
        if(MaHumo){
            if(detector_humo() == 1 && bocina == 0 && band_h){
                band_h = false;
                System.out.println("ENCIENDE BOCINA");
                tiempo(600);
                enviadato("H");
            }
            if(detector_humo() == 0){
                band_h=true;
            }
        }
        if(MaGas){
            if(detector_gas() == 1 && bocina == 0 && band_g){
                band_g = false;
```



```

        System.out.println("ENCIENDE BOCINA");
        tiempo(600);
        enviadato("H");
    }
    if(detector_humo() == 0){
        band_g=true;
    }
}
if(MaMagnetico){
    if(sensor_magnetico1() == 1 && bocina == 0 && band_m1){
        band_m1 = false;
        System.out.println("ENCIENDE BOCINA");
        tiempo(600);
        enviadato("H");
    }
    if(sensor_magnetico1() == 0){
        band_m1=true;
    }
    if(sensor_magnetico2() == 1 && bocina == 0 && band_m2){
        band_m2 = false;
        System.out.println("ENCIENDE BOCINA");
        tiempo(600);
        enviadato("H");
    }
    if(sensor_magnetico2() == 0){
        band_m2=true;
    }
}
}
}
}

```

Si el fluido eléctrico llegara a fallar en cualquiera de los módulos de comunicaciones, el evento es informado al servidor de forma inmediata y este notifica mediante correo o SMS lo acontecido.

Interpreta los datos recibidos por los módulos de comunicaciones.

```

if (estado.equalsIgnoreCase("CorrectoX1")){
    xbee1 = 1 ;
    historial_xbee();
}
if (estado.equalsIgnoreCase("FallaX1")){
    xbee1 = 2 ;
    historial_xbee();
}

```

Guardan en el historial lo acontecido y notifican al usuario vía correo electrónico.

```

public void historial_xbee(){
    obtener_fecha();
    if(xbee1==1 && inicializacion){
        cbdd.historial_xbee("1", "Energia xbee 1 correcta", fecha, hora);
        enviar_mail(" Notificación Detector de Energia ", "El circuito Xbee1
tiene un correcto fluido de energía \n Energía correcta\n Fecha :
"+fecha+"\n Hora : "+hora);
    }
    if(xbee1==2 && inicializacion){
        cbdd.historial_xbee("1", "Energia xbee 1 Falla", fecha, hora);
        enviar_mail(" Notificación Detector de Energia ", "El circuito Xbee1
sufrió un corte de energía \nFallo de energía\n Fecha : "+fecha+"\n
Hora : "+hora);
    }
}

```

El módulo GLCD tiene un botón de pánico denominado SOS o emergencia, cuando es presionado inmediatamente es interpretado por el servidor y envía una notificación vía correo electrónico para que se tomen las debidas precauciones.

```

if (estado.equalsIgnoreCase("EMERGENCIA")){
    emergencia = 0;
    tiempo(4000);
    emergencia = 1;
    codmedio = "3";
    System.out.println("EMERGENCIA");
}

```

```

tipo_usu();
estvariable="EMERGENCIA";
insertar_histo();
enviar_mail(" Notificación EMERGENCIA ", "Llamada de emergencia
\n Fecha : "+fecha+"\n Hora : "+hora+"\n Por medio del servicio
"+medio+" \n Comprobar novedades...");
estado = "";
}

```

#### 5.3.3.2.4. Módulo Servidor Correo

Se crea una nueva clase llamada EnviarCorreo la cual será llamada cuando el servidor requiera enviar notificaciones vía correo electrónico.

Esta clase ocupa dos librerías adicionales que no vienen incluidas en Java por tal motivo deben ser descargadas de la web e insertarlas en Java. Estas dos librerías son JavaMail y JAF (JavaBeans Activation Framework).

A continuación, se mostrará como agregar estas librerías a Netbeans:

- 1.- Se presiona botón secundario sobre la aplicación Web y seleccionar la opción Propiedades tal como se muestra en la siguiente figura.

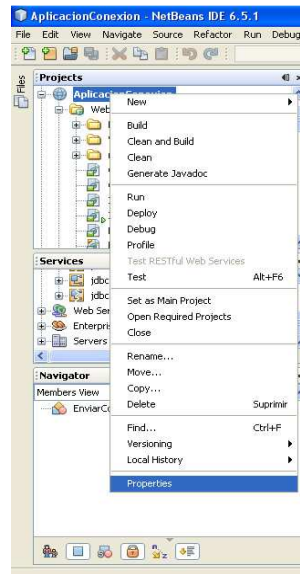


Figura 174: Como agregar librerías en Netbeans Paso 1

2.- Luego aparece una nueva pantalla en la cual se escoge la opción Libraries, y se muestran todas la librerías con las que se está trabajando. En esta pantalla se presiona sobre el botón Add JAR\Folder como se muestra en la siguiente figura.

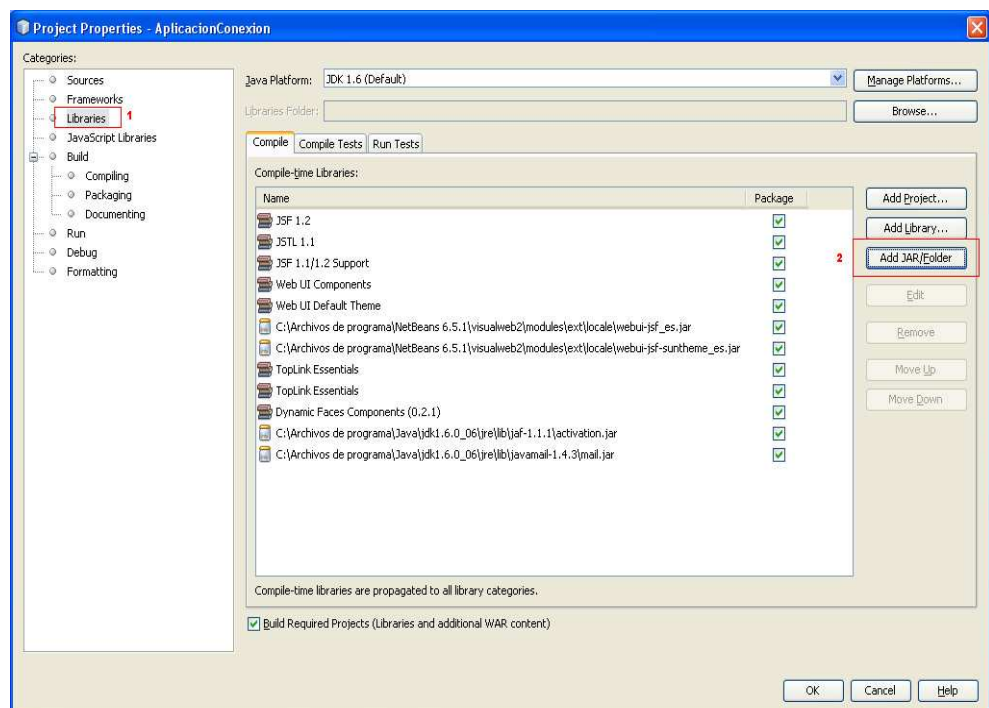


Figura 175: Como agregar librerías en Netbeans Paso 2

3.- Al presionar sobre el botón Add JAR\Folder se despliega la ventana para buscar las 2 librerías que se necesita agregar, la una es mail.jar y activation.jar

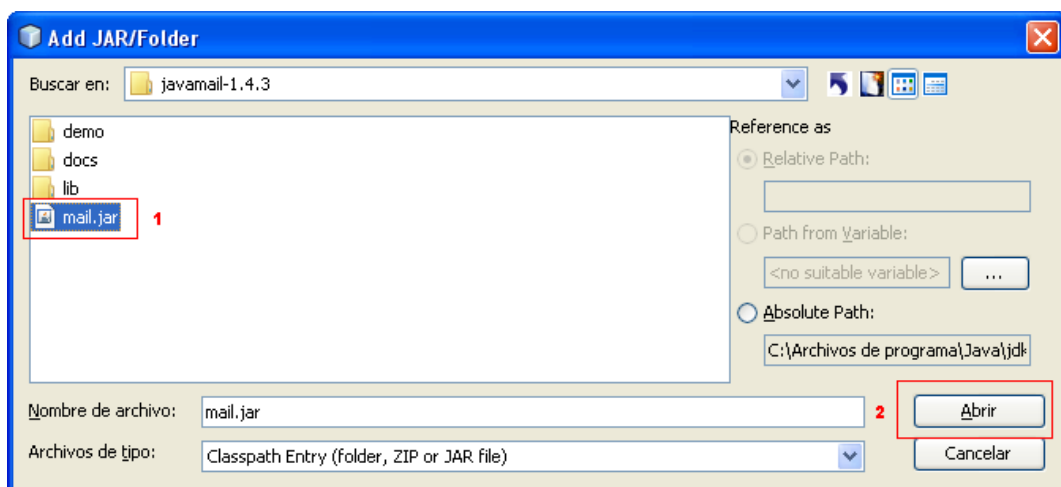


Figura 176: Como agregar librerías en NetBeans Paso 3

4.- Se realiza el mismo procedimiento del paso 3 para agregar la otra librería activation.jar y de esta forma ya se tiene las nuevas librerías lista para ser usadas.

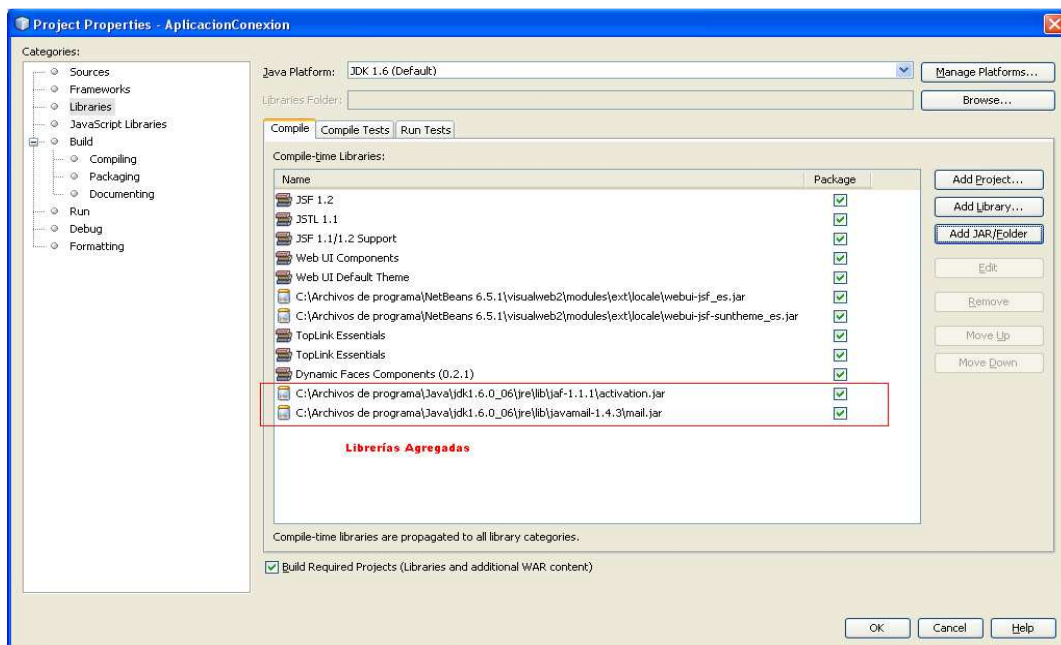


Figura 177: Como agregar librerías en NetBeans Paso 4

Ya agregadas las librerías se sigue con el desarrollo de la clase `EnviarCorreo`, al inicio del código se importa lo siguiente:

```
import javax.mail.*;
import javax.mail.internet.*;
import java.util.*;
```

Se requieren de tres clases de la librería `JavaMail` que son:

- **Session:** Representa la conexión con el servidor gmail de correo.
- **Transport:** Clase para el envío de mensajes. Se obtiene llamando al método `getTransport()` de la clase `Session`.
- **MimeMessage:** El mensaje que se va a enviar.

Mediante el siguiente código generado se observa el uso de las tres clases señaladas anteriormente las cuales sirven para enviar correos.

Se creó una cuenta llamada `zigbeedomotica@gmail.com` por la cual se enviarán los correos de alerta y notificación. Es posible enviar tres copias a distintos correos (TO, CC, BCC) como se ve en el siguiente código, todos estos parámetros son enviados desde las clases del servidor y son consultadas de la base de datos de la tabla `Usuarios`.

```
try
{
    Session session = Session.getInstance(props, null);
    session.setDebug(debug);
    MimeMessage msg = new MimeMessage(session);
    msg.setText(text);
    msg.setSubject(subject);
    msg.setFrom(new
InternetAddress("zigbeedomotica@gmail.com"));
```

```

        msg.addRecipient(Message.RecipientType.TO, new
InternetAddress(to));
        msg.addRecipient(Message.RecipientType.CC, new
InternetAddress(cc));
        msg.addRecipient(Message.RecipientType.BCC, new
InternetAddress(bcc));
        msg.saveChanges();
        Transport transport = session.getTransport("smtp");
        transport.connect("smtp.gmail.com",
"zigbeedomotica@gmail.com", "tesisups");
        transport.sendMessage(msg, msg.getAllRecipients());
        transport.close();
        return true;
    }
    catch (Exception mex)
    {
        mex.printStackTrace();
        return false;
    }
}

```

#### 5.3.3.2.5. Módulo Servidor Cámara IP

El siguiente módulo se generó de manera muy intuitiva y sencilla. Al ya contar con un software especial la cámara IP solo basto con recurrir a ése código e implementarlo al proyecto, es así que se inserta dentro del código JSP creado por Netbeans dentro del formulario que le corresponde a Camara IP.

El código insertado es un Applet que contiene parámetros de inicio de la cámara tales como dirección IP, el tamaño de la ventana, el puerto etc., que ayudaran a definir la configuración que regirá la cámara.

El siguiente fragmento de código, se debe insertar para que la cámara IP se muestre en el servidor web.

```

<APPLET CODE="xplug.class" CODEBASE="http://10.1.1.3:80/"
name="cvcs" style="height: 47%; left: 25%; top: 20%; position:absolute;
width: 50%">

    <param name="RemotePort" value="80"/>

    <param name="Timeout" value="5000"/>

    <param name="RotateAngle" value="0"/>

    <param name="PreviewFrameRate" value="2"/>

    <param name="DeviceSerialNo"
value="YWRtaW46"/>

</APPLET>

```

También se agregó 4 botones que generan un Zoom a las imágenes que envía la cámara IP, esto se observa en el siguiente código.

```

<input onClick="cvcs.Zoom(1)" style="height: 24px; left: 168px; top:
384px; position: absolute; width: 45px" type="button" value="x1"/>

<input onClick="cvcs.Zoom(2)" style="height: 24px; left: 240px; top:
384px; position: absolute; width: 45px" type="button" value="x2"/>

<input onClick="cvcs.Zoom(3)" style="height: 24px; left: 312px; top:
384px; position: absolute; width: 45px" type="button" value="x3"/>

<input onClick="cvcs.Zoom(4)" style="height: 24px; left: 384px; top:
384px; position: absolute; width: 45px" type="button" value="x4"/>

```

En la siguiente figura, se muestra como se visualiza imágenes enviadas por la cámara IP hacia el servidor web.



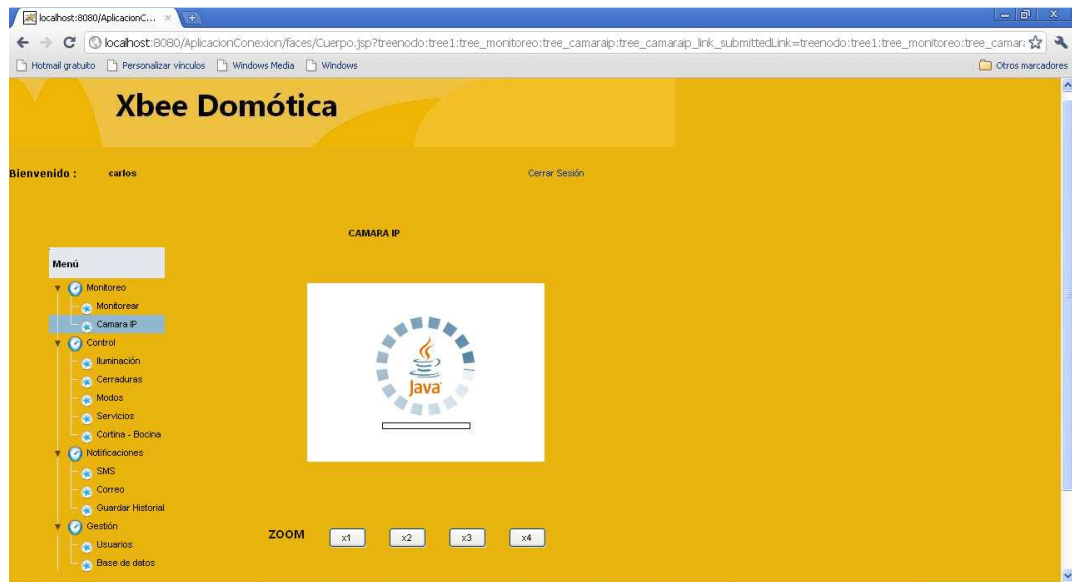


Figura 178: Cámara IP Mostrada en el servidor Web

### 5.3.3.3. Interfaces

Las interfaces del servidor usan los siguientes objetos instanciados para sus diferentes funciones.

```

Coneccion cxn = new Coneccion();
clases.XbeeTouch xbt = new clases.XbeeTouch();
clases.Bluetooth bt = new clases.Bluetooth();
clases.smsx smsx = new clases.smsx();
clases.HiloLeeSms leesms = new clases.HiloLeeSms();
clases.HiloMonitoreo monitosms = new clases.HiloMonitoreo();
clases.consultasbd consul = new clases.consultasbd ();
clases.HiloMSP HiloMSP = new clases.HiloMSP();
clases.Validaciones valid = new clases.Validaciones();
UserController userControllerx = new UserController();
  
```

#### 5.3.3.3.1. Inicio de sesión

La figura 179 muestra la interfaz de inicio de sesión, aquí se ingresa el nombre de usuario, contraseña y tipo de usuario. Al presionar el botón iniciar sesión se realiza la comprobación con la base de datos si existe un usuario con estos datos, se inicia la sesión, se asignan las variables globales de sesión con el nombre de usuario y contraseña.

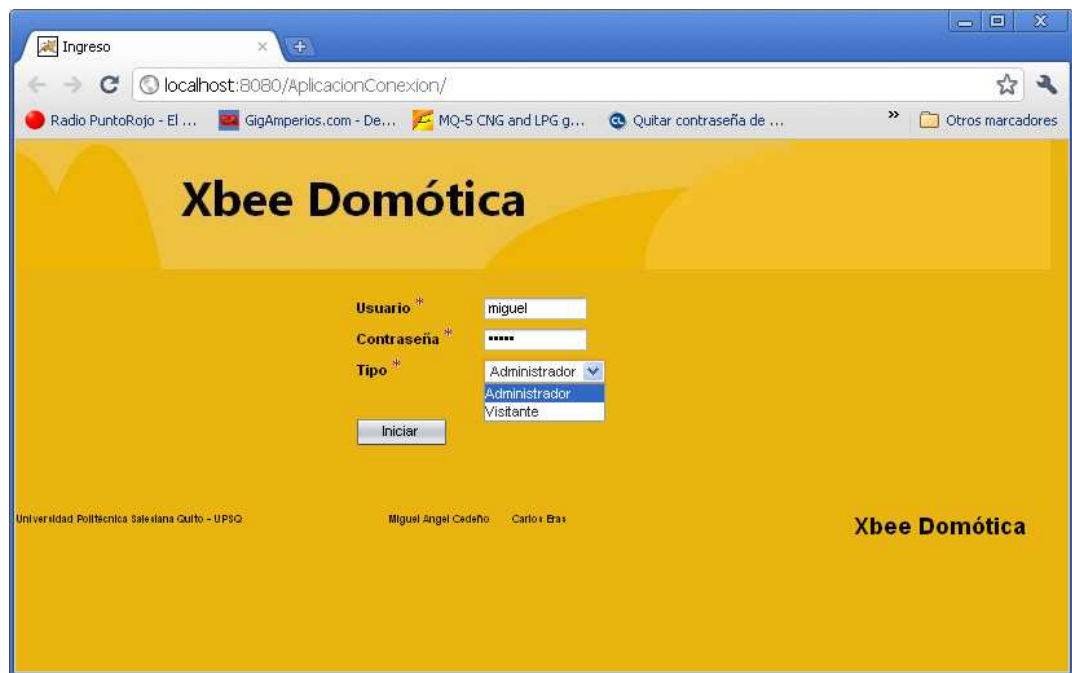


Figura 179: Interfaz de inicio de sesión

Fragmento de código, del botón iniciar, aquí se muestra como se realiza la verificación del usuario con la base de datos.

```
public String buttonLogin_action() {  
    getSessionBean1().setUsuario((String)this.txtusuario1.getText());  
    getSessionBean1().setContraseña((String)this.txtcontrasena1.getText());  
    getSessionBean1().setTipo_usuario((String)this.cbxtipo1.getSelected().toString()  
    );  
    String t=this.cbxtipo1.getSelected().toString();  
    String u=txtusuario1.getText().toString();
```

```

String c=txtcontrasena1.getText().toString();
try
{
    reg = cxn.Consulta("SELECT * FROM usuarios where nombre='"+u+"'
and contra='"+c+"' and cod_tipo='"+t+"'");
    if (reg.next()){
        System.out.println("Usuario correcto");
        return "correcto";
    }
    else {
        System.out.println("no existe el usuario");
        return "incorrecto";
    }
}
}catch(SQLException ex){System.out.println(ex);}
cxn.Ejecutar("drop view guarda");
return "correcto" ;
}
}

```

Si el usuario es incorrecto se muestra la siguiente interfaz.



Figura 180: Interfaz Usuario Incorrecto

Si el usuario es correcto, se muestra la siguiente interfaz de bienvenida.

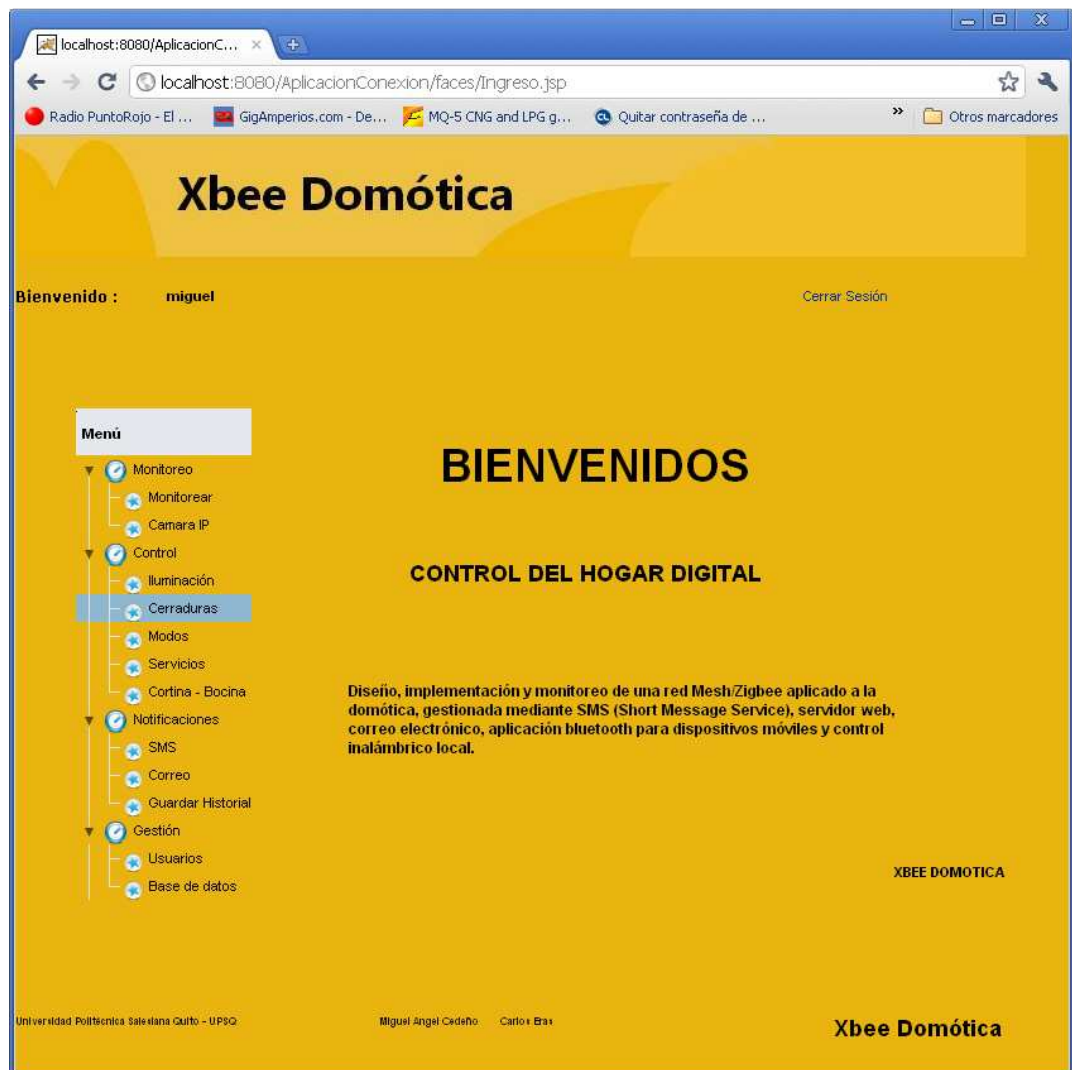


Figura 181: Interfaz Usuario Incorrecto

Para cerrar la sesión clic en el botón “cerrar sesión” ubicado en la parte superior derecha, y se muestra la siguiente interfaz.



Figura 182: Interfaz Cerrar sesión

Al ingresar con un usuario de tipo “Invitado” están restringidas las opciones de administración y el control de las cerraduras, el menú principal que se despliega es: ver figura 183.

Al ingresar con un usuario de tipo “Administrador” están habilitadas todas las opciones, el menú principal que se despliega es: ver figura 184.



Figura 183: Menú tipo Invitado



Figura 184: Menú tipo Administrador

En la siguiente figura, se muestra las páginas de navegación o flujo de página que se diseñó para la implementación de sesiones

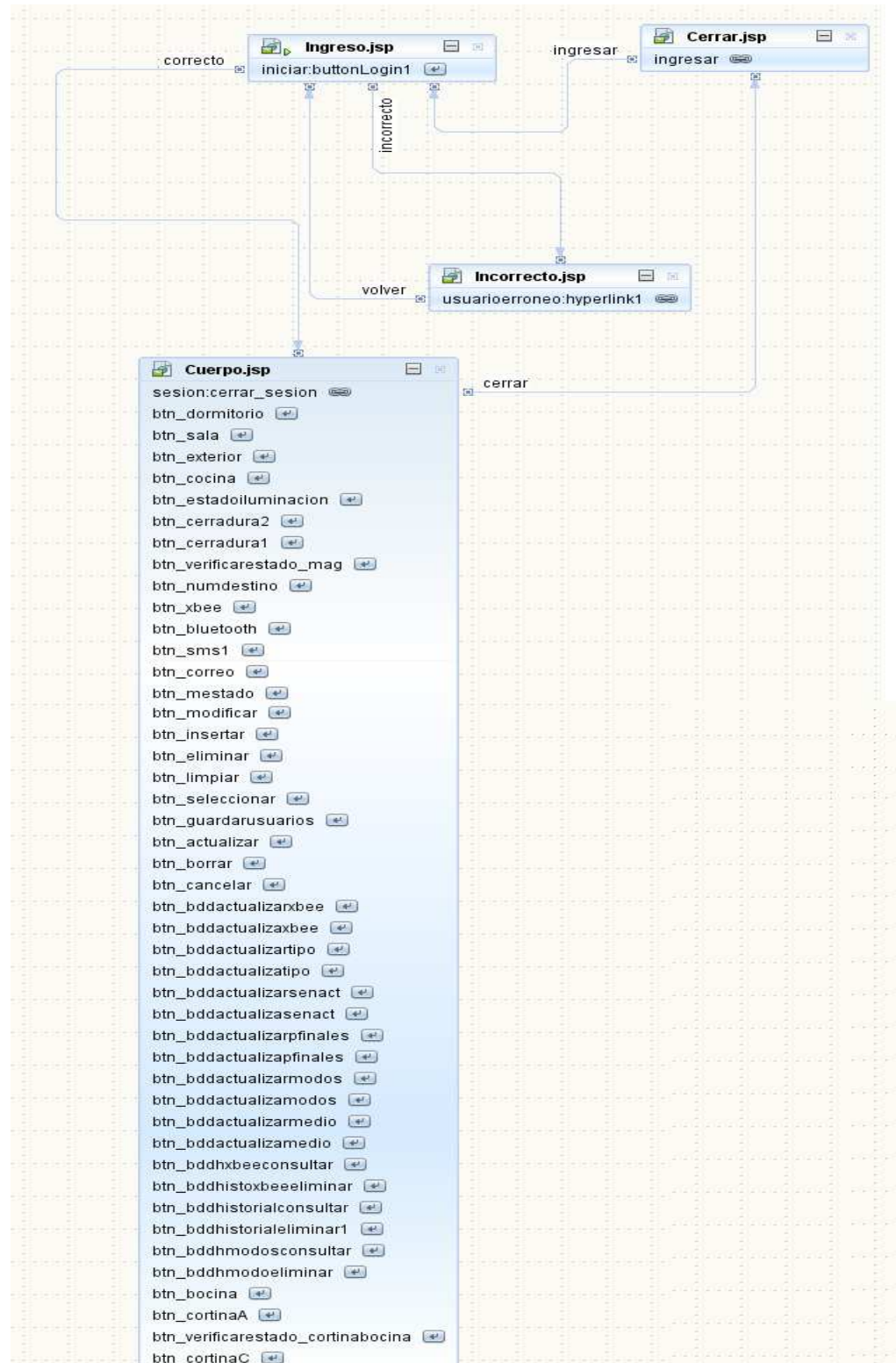


Figura 185: Páginas de navegación o Flujo de Página

#### **5.3.3.3.2. Monitorear**

La interfaz monitorear se muestra en la siguiente figura, aquí se puede observar el estado de la totalidad de los dispositivos finales sensores y actuadores.

La consulta se realiza al módulo servidor Zigbee y este devuelve el estado actual de los dispositivos finales, cuando están activados se muestra con un color azul y si están desactivados con un color rojo.

La parte de Módulos de comunicaciones muestran la información acerca del estado del fluido eléctrico en cada uno de ellos, si está bien muestra “Correcto” y si el fluido eléctrico falla muestra “Falla de energía”

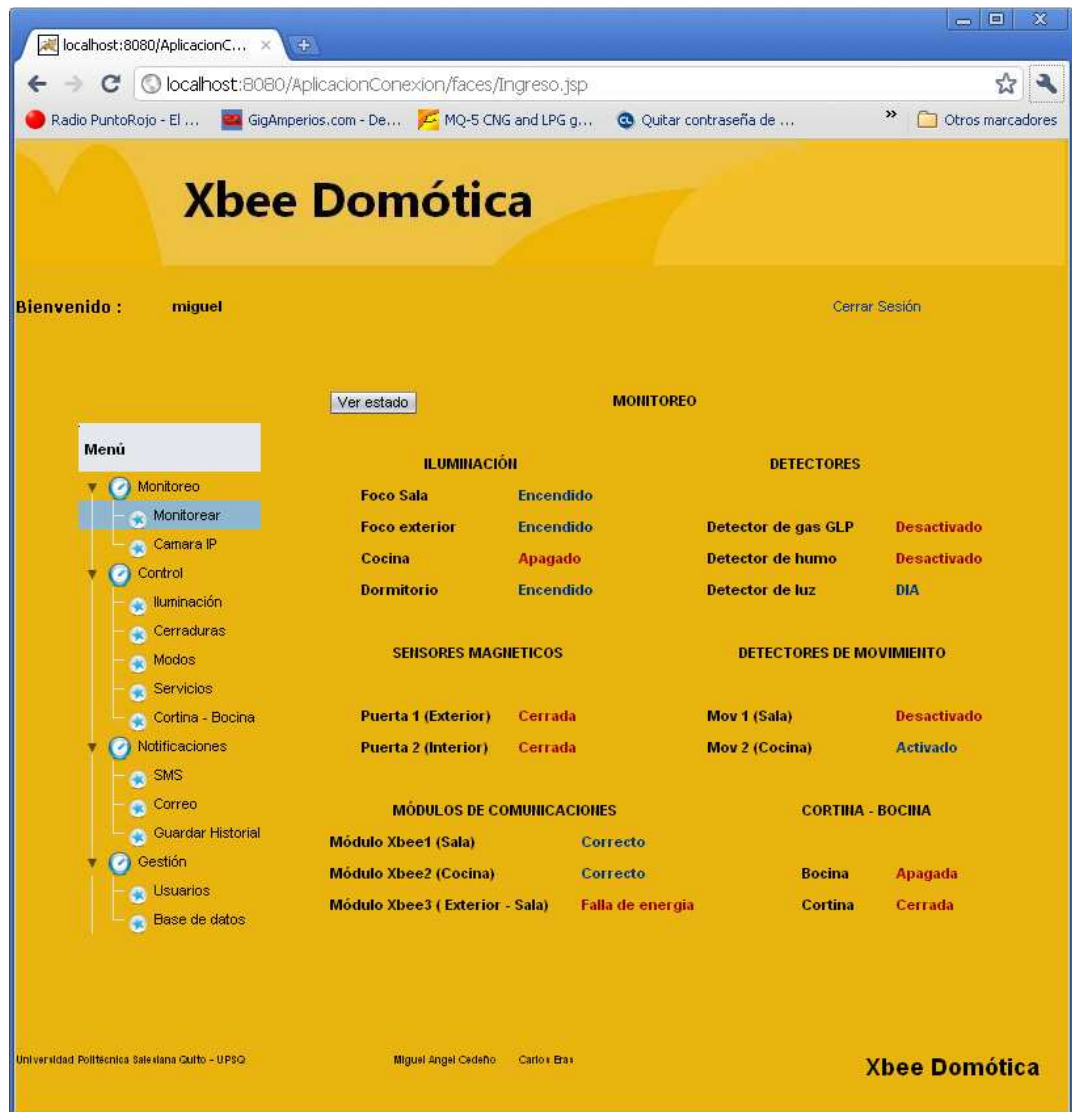


Figura 186: Interfaz Monitorear

El botón ver historial devuelve el estado actual de cada uno de los dispositivos finales cuando se necesite.

Este fragmento de código, muestra como se realiza la petición de estados al módulo servidor Zigbee y luego se imprime en el formulario con los respectivos colores para la correcta visualización.



```

public void monitorear(){
    //iluminacion
    if
(xbt.foco_sala()==1){this.lbl_msala.setText("Encendido");lbl_msala.setStyle("col
or: #003366; font-weight: bold; left: 168px; top: 96px; position: absolute");}
    if
(xbt.foco_sala()==0){this.lbl_msala.setText("Apagado");lbl_msala.setStyle("colo
r: #990000;font-weight: bold; left: 168px; top: 96px; position: absolute");}
    if
(xbt.foco_exterior()==1){this.lbl_mexterior.setText("Encendido");lbl_mexterior.s
etStyle("color: #003366;font-weight: bold; left: 168px; top: 120px; position:
absolute");}
    if
(xbt.foco_exterior()==0){this.lbl_mexterior.setText("Apagado");lbl_mexterior.set
Style("color: #990000;font-weight: bold; left: 168px; top: 120px; position:
absolute");}
    if
(xbt.foco_cocina()==1){this.lbl_mcocina.setText("Encendido");lbl_mcocina.setSt
yle("color: #003366;font-weight: bold; left: 168px; top: 144px; position:
absolute");}
    if
(xbt.foco_cocina()==0){this.lbl_mcocina.setText("Apagado");lbl_mcocina.setStyl
e("color: #990000;font-weight: bold; left: 168px; top: 144px; position:
absolute");}
    if
(xbt.foco_dormitorio()==1){this.lbl_mdormitorio.setText("Encendido");lbl_mdorm
itorio.setStyle("color: #003366;font-weight: bold; left: 168px; top: 168px;
position: absolute");}
    if
(xbt.foco_dormitorio()==0){this.lbl_mdormitorio.setText("Apagado");lbl_mdormit
orio.setStyle("color: #990000;font-weight: bold; left: 168px; top: 168px; position:
absolute");}

```

### 5.3.3.3. Control Iluminación

En la siguiente figura, se muestra la interfaz control de iluminación, aquí es posible controlar los diferentes focos del hogar encenderlos y apagarlos, con solo dar un clic en el botón correspondiente.

También es posible monitorear el estado de los focos en la parte derecha por medio del botón Verificar estado.

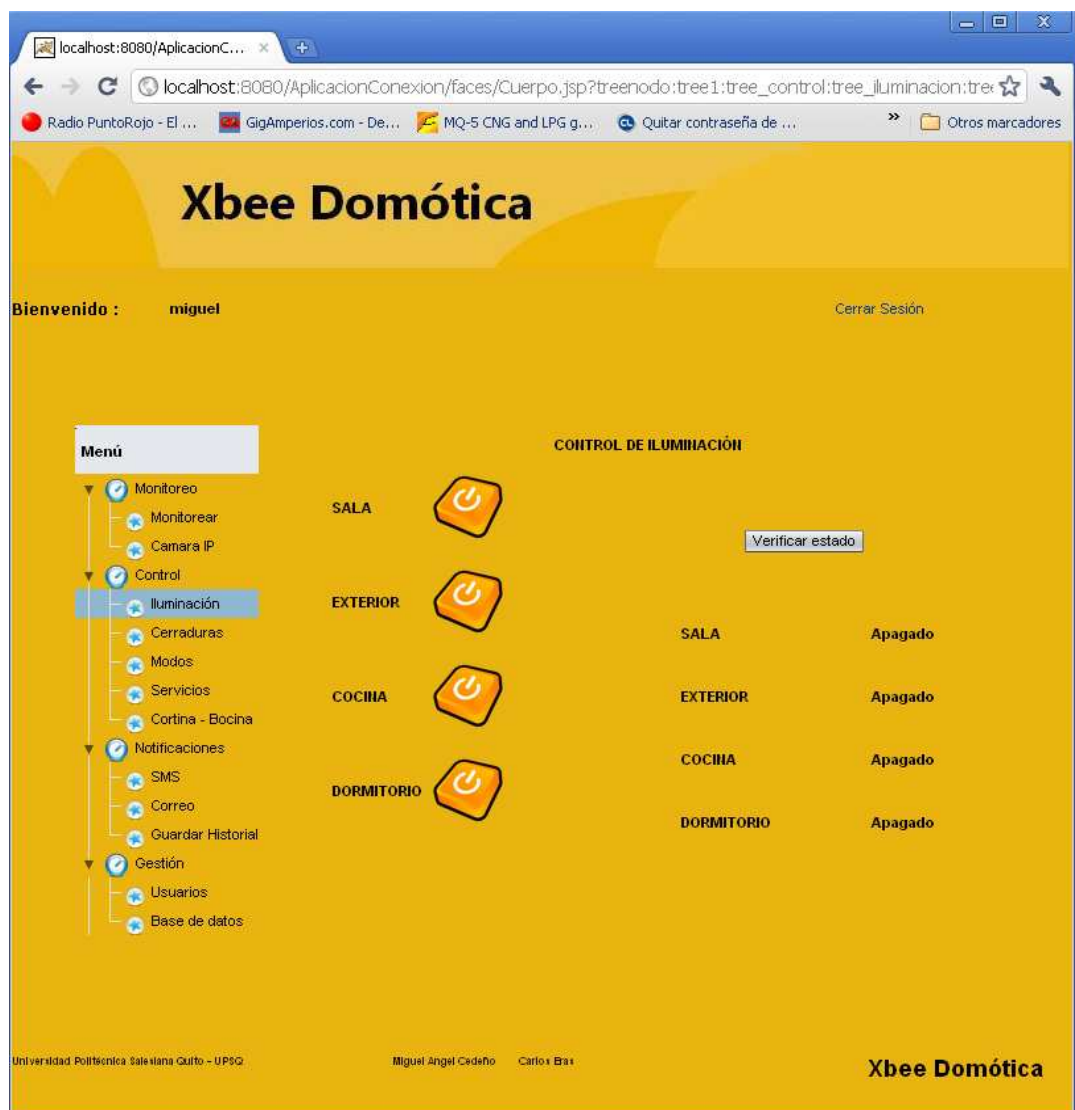


Figura 187: Interfaz Control de Iluminación

Para evitar posibles inconvenientes al enviar muy seguidos los códigos de comunicaciones se usa un timer, su construcción es sencilla con la ayuda de un hilo.

```
public void tiempo(int t) {  
    try {  
        Thread.sleep(t);  
    } catch (Exception e) {  
    }  
}
```

Este fragmento de código, muestra como un botón realiza la petición de cambio de estado al módulo servidor Zigbee, para que por medio de él se envíe el código de comunicación para encender o apagar un foco.

```
public String btn_cocina_action() {  
    tiempo(1000);  
    xbt.enviadoato("C");  
    xbt.usuario_web(this.lbl_codigouser.getText().toString());  
    estado_iluminacion();  
    return null;  
}
```

De igual manera se programan todos botones para el envío de códigos de comunicaciones de encendido o apagado.

#### **5.3.3.3.4. Control Cerraduras**

En la siguiente figura, se muestra la interfaz control de cerraduras, está restringida solo para usuarios de tipo “Administrador”, debido a su sensibilidad en el hogar se toman ciertas medidas de seguridad para evitar que cualquier usuario active una cerradura y pueda ingresar sin autorización.

Esta interfaz cuenta con dos botones uno para cada cerradura ubicada en cada puerta, al hacer clic sobre ellos se activa la cerradura permitiendo que la puerta se abra. También se cuenta

con un pequeño monitoreo de los sensores magnéticos ubicados en estas puertas con el fin de verificar si son cerradas luego de esta acción.

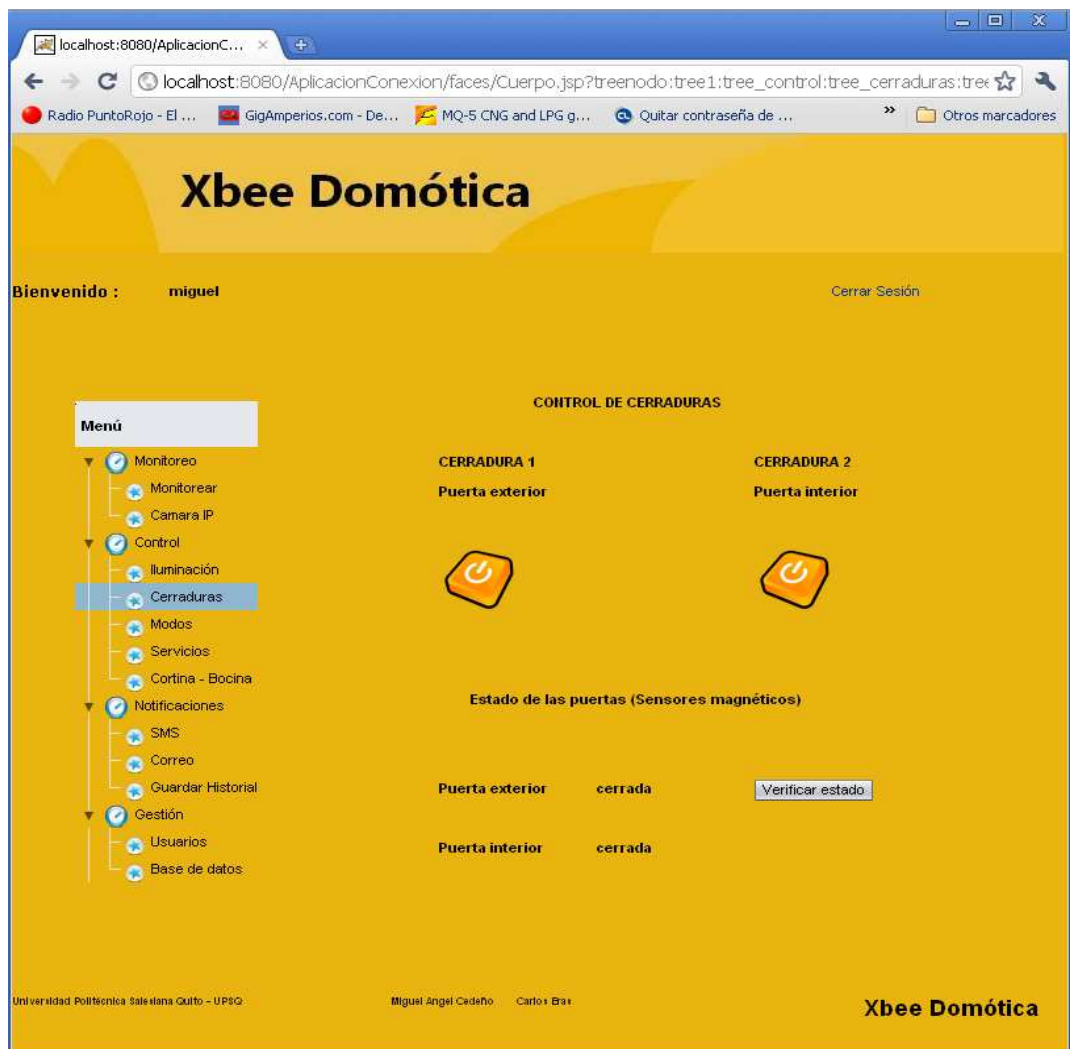


Figura 188: Interfaz Control de Cerraduras

Este fragmento de código, muestra como un botón realiza la petición de activar cerradura al módulo servidor Zigbee ,para que por medio de él se envíe el código de comunicación.

```
public String btn_cerradura1_action() {
    tiempo(1000);
    xbt.enviadato("F");
    xbt.usuario_web(this.lbl_codigouser.getText().toString());
    tiempo(2000);
    return null;
}
```

Para el monitoreo de los sensores magnéticos ubicados en las puertas se llama a un método verificar estado que monitorea a estos sensores.

```
public String btn_verificareestado_mag_action() {
    estado_magneticos();
    return null;
}

public void estado_magneticos() {
    int mag1;
    int mag2 ;
    mag1=xbt.sensor_magnetico1();
    mag2=xbt.sensor_magnetico2();
    if (mag1==1){this.lbl_sensormag1.setText("Abierta");}
    if (mag1==0){this.lbl_sensormag1.setText("cerrada");}
    if (mag2==1){this.lbl_sensormag2.setText("Abierta");}
    if (mag2==0){this.lbl_sensormag2.setText("cerrada");}
}
```

#### **5.3.3.3.5. Control Cortina – Bocina**

En la siguiente figura, se muestra la interfaz control de cortina y bocina, no está restringida para ningún tipo de usuario, aquí se puede abrir o cerrar la cortina y encender o apagar la bocina.

Esta interfaz cuenta con 3 botones, 2 para el control de apertura / cierre de la cortina y 1 para encender / apagar la bocina.

También cuenta con un pequeño monitoreo que muestra el estado actual de la cortina si está abierta o cerrada y el estado actual de la bocina.



Figura 189: Interfaz Control de Cortina y Bocina

Este fragmento de código, muestra como un botón realiza la petición de abrir cortina al módulo servidor Zigbee, por medio de él se envíe el código de comunicación.

```
public String btn_abrirrc_action() {
    tiempo(1000);
    xbt.enviadato("J");
    xbt.usuario_web(this.lbl_codigouser.getText().toString());
    tiempo(2000);
    return null;
}
```

El monitoreo se realiza haciendo peticiones de estado al presionar el botón “Verificar Estado” al igual que en los monitoreos anteriores.

#### 5.3.3.3.6. Control Modos

En la siguiente figura, se muestra la interfaz control de modos, está restringida solo para usuarios de tipo “Administrador”, Aquí es posible controlar los 3 tipos de modos que cuenta el sistema.

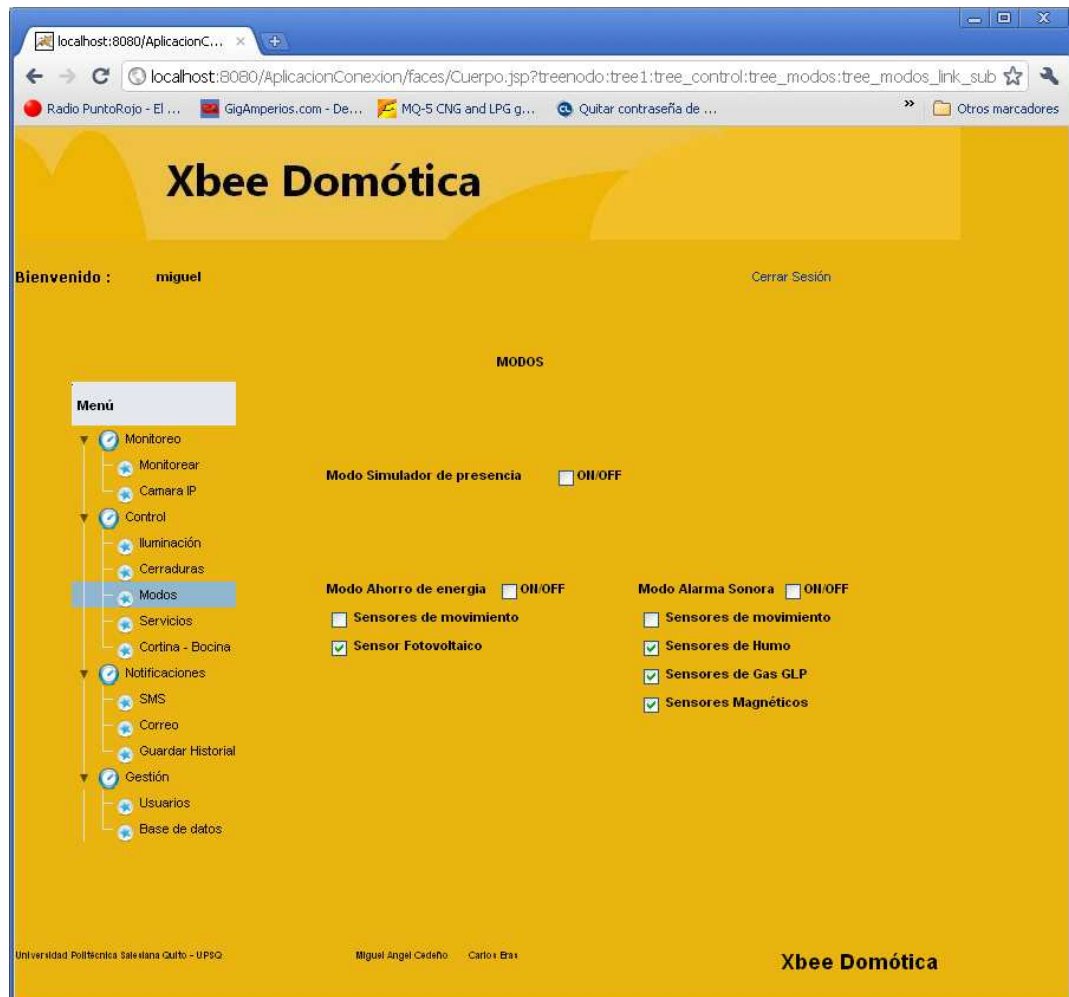


Figura 190: Interfaz Control Modos

#### Modo simulador de presencia

Con la ayuda de un checkbox es posible activarlo o desactivarlo, si se activa inicia un hilo que controla este modo, si se desactiva para este hilo.

```

if (chkMSP.isChecked()){
    xbt.mp(1);
    System.out.println("MP ON");
    HiloMSP.start();
    HiloMSP.asigna(true);
}
else
{
    xbt.mp(0);
    System.out.println("MP OFF");
    HiloMSP.asigna(false);
}

```

### **Hilo Modo simulador de presencia**

La necesidad de crear un hilo para realizar este proceso radica en que el servidor debe seguir funcionando a pesar de que se contabilicen los tiempos para activar o desactivar actuadores, este proceso debe ser paralelo al servidor.

Se crea el hilo y se instancia a la clase “XbeeTouch” por medio de ella se envían los códigos de comunicaciones.

```

public class HiloMSP extends Thread{
    clases.XbeeTouch xbt = new clases.XbeeTouch();
    static boolean MSP;
    static String dato;
    static int tiempor;
    static double rnd;
    @Override
    public void run()
    {
        MSP = true;
        while(MSP){
            randomico ();
            xbt.enviadato(dato);
            tiempo(tiempor);
        }
    }
}

```



La manera ideal para que este modo simulador de presencia sea efectivo y en realidad simule la presencia de personas en el interior del hogar, es que los eventos que realice de encender o apagar actuadores sea de manera aleatoria y en tiempos aleatorios, para esto se usa un número randómico como se muestra en el siguiente fragmento de código.

```
public void randomico (){
    rnd = Math.random() * 10;
    int rndr = (int) rnd;
    tiempor = rndr*10000;
    if(rndr < 2){
        dato="A";
    }
    if(rndr >= 2 && rndr < 4 ){
        dato="B";
    }
    if(rndr >= 4 && rndr < 5 ){
        dato="I";
    }
    if(rndr >= 5 && rndr < 6 ){
        dato="J";
    }
    if(rndr >= 6 && rndr < 8 ){
        dato="C";
    }
    if(rndr >= 8 && rndr < 10 ){
        dato="D";
    }
}
```

### **Modo Ahorro de energía**

Con la ayuda de un chekbox es posible activar o desactivar este modo, funciona bajo dos criterios de ahorro, los detectores de movimiento y el sensor foto voltaico.

Cuando no detecta movimiento en la habitación apaga los focos y cuando detecta los enciende, si el sensor fotovoltaico detecta que Luz natural apaga el foco exterior si no detecta Luz lo enciende.

Este modo no funciona con hilos, simplemente está incluido en el módulo Servidor Zigbee, solo controla los criterios de ahorro.

En el siguiente fragmento de código, se muestra como se activa o se desactiva este modo bajo, los criterios de ahorro de energía usando checkbox.

```
public void setear_MAE(){
    if (chkMAE.isChecked()){
        xbt.mae(1);
        System.out.println("MAE ON");
        if (chk_movMae.isChecked()){
            xbt.mae_mov(true);
        }
        else{
            xbt.mae_mov(false);
        }
        if (chk_fotoMae.isChecked()){
            xbt.mae_fotov(true);
        }
        else{
            xbt.mae_fotov(false);
        }
    }
    else
    {
        xbt.mae(0);
        System.out.println("MAE OFF");
    }
}
```

## Modo Alarma Sonora

Este modo consiste en activar o desactivar la bocina cuando se active o desactive algún sensor, tales como detectores de movimiento, humo gas y sensores magnéticos.

Para seleccionar la condición del modo alarma sonora se usan checkbox en el siguiente fragmento de código, se muestra como.

```
public void setear_MA(){
    if (chk_MA.isChecked()){
        xbt.ma(1);
        System.out.println("MA ON");
        if (chk_movMa.isChecked()){
            xbt.ma_mov(true);
        }
        else{
            xbt.ma_mov(false);
        }
        if (chk_humoMA.isChecked()){
            xbt.ma_humo(true);
        }
        else{
            xbt.ma_humo(false);
        }
        if (chk_gasMA.isChecked()){
            xbt.ma_gas(true);
        }
        else{
            xbt.ma_gas(false);
        }
        if (chk_magneticoMA.isChecked()){
            xbt.ma_magnetico(true);
        }
        else{
            xbt.ma_magnetico(false);
        }
    }
}
```

### 5.3.3.3.7. Servicios

En la siguiente figura, se muestra la interfaz Iniciar servicios esta sirve para iniciar los módulos servidores cuando el servidor principal haya sido reiniciado.

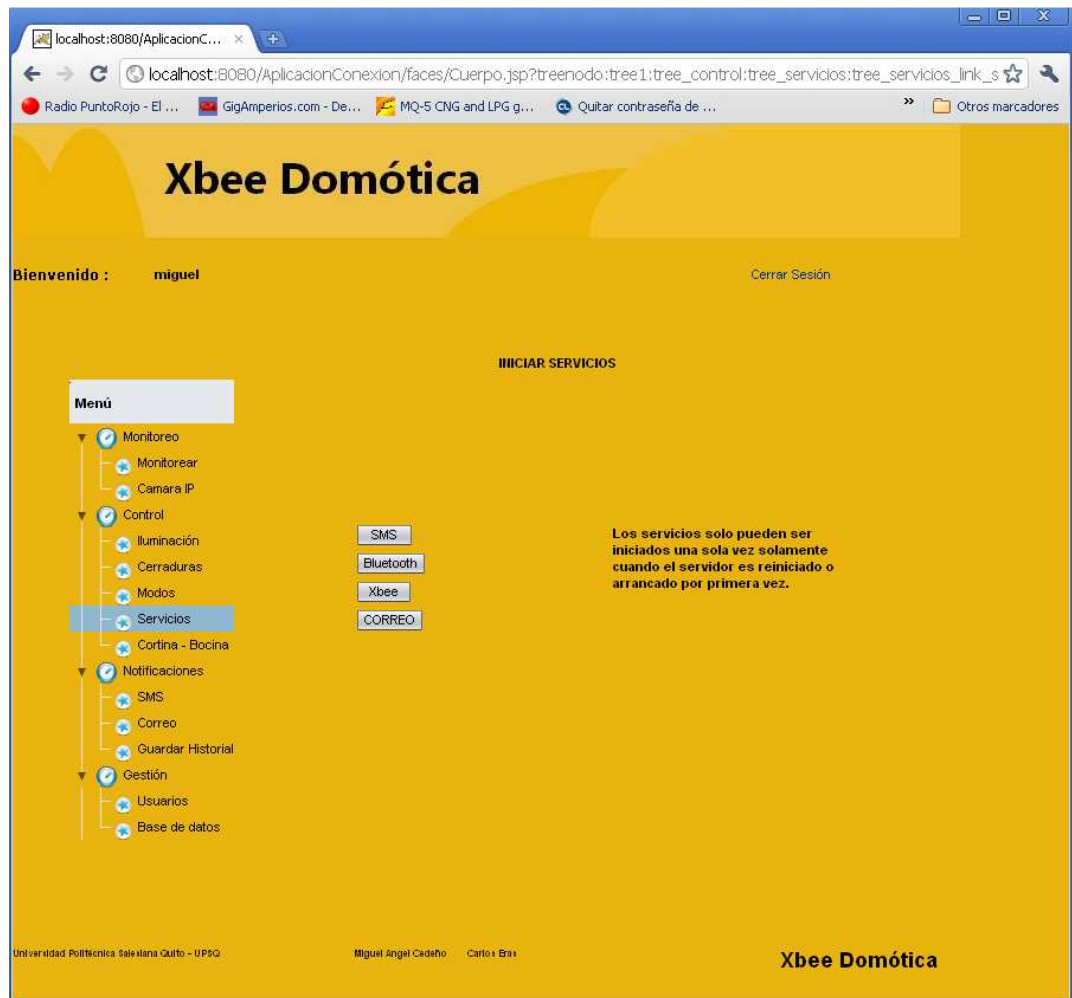


Figura 191: Interfaz Iniciar Servicios

Cuenta con 4 botones cada uno representa a un módulo servidor a ser inicializado.

Botón SMS inicia el modem GSM y también el hilo de lectura SMS.

```

public String btn_sms1_action() {

    smsx.start();
    smsx.enviar("AT+CMGF=1");
    leesms.start();
    String cel_dest ;
    cel_dest = "\"" + txt_celdestino.getText().toString() + "\"";
    smsx.asignar_número(cel_dest);
    monitosms.start();
    if(getSessionBean1().getUsuario().length()>0 ){
        btn_sms1.setVisible(false);
    }
    return null;
}

```

Botón Bluetooth inicia el hilo Bluetooth para que escuche el puerto COM.

```

public String btn_bluetooth_action() {
    bt.start();
    bt.enviadata(" ");
    if(getSessionBean1().getUsuario().length()>0 ){
        btn_bluetooth.setVisible(false);
    }
    return null;
}

```

Botón Xbee inicia el hilo XbeeTouch y también inicializa los módulos de comunicaciones Xbee1, Xbee2 y Xbee3 pidiendo el estado actual de sus dispositivos finales.

```

public String btn_xbee_action() {
    xbt.start();
    setearmodos_grabar();
    if(getSessionBean1().getUsuario().length()>0 ){
        btn_xbee.setVisible(false);
    }
    xbt.xbee1();
}

```

```

tiempo(6000);
xbt.xbee2();
tiempo(6000);
xbt.xbee3();
tiempo(6000);
xbt.xbee2final();
return null;
}

```

### 5.3.3.3.8. Notificaciones SMS

En la siguiente figura, se muestra la interfaz de Notificaciones SMS, la cual permite configurar bajo qué eventos se envían las notificaciones al usuario, está restringida solo para el uso del usuario de tipo “Administrador” .



Figura 192: Interfaz Notificaciones SMS

Consta de una caja de texto donde el usuario administrador ingresa el número de celular al cual necesita que se le envíen este tipo de notificaciones. Al hacer clic en el botón “Cambiar” se asigna el nuevo número que se ingresa en el módulo servidor SMS, a continuación, se muestra un fragmento de código, del botón “Cambiar”.

```
public String btn_numdestino_action() {  
    String cel_dest ;  
    cel_dest = "\"" + txt_celdestino.getText().toString() + "\"";  
    smsx.asignar_número(cel_dest);  
    return null;  
}
```

Se pueden activar o desactivar los eventos bajo los cuales se envíen este tipo de notificaciones, en el siguiente fragmento de código, se muestra como se manejan los checkbox de esta interfaz.

```
public void monitoreosms(){  
    if (chkMonitoreoSMS.isChecked()){  
        smsx.start();  
        smsx.todo(true);  
        System.out.println("Monitoreo SMS activado");  
        //Monitoreo iluminacion  
        if(chk_iluminacionSMS.isChecked()){  
            this.smsx.iluminacion(true);  
        }  
        else  
        {  
            this.smsx.iluminacion(false);  
        }  
        //Monitoreo movimiento  
        if(chk_movSMS.isChecked()){  
            this.smsx.movimiento(true);  
        }  
        else  
        {
```

```

        this.smsx.movimiento(false);
    }
    //Monitoreo gas
    if(chk_humoSMS.isChecked()){
        this.smsx.humo(true);
    }
    else
    {
        this.smsx.humo(false);
    }
    //Monitoreo humo
    if(chk_gasSMS.isChecked()){
        this.smsx.gas(true);
    }
    else
    {
        this.smsx.gas(false);
    }
    // Monitoreo Magnetico
    if(chk_magneticoSMS.isChecked()){
        this.smsx.magnetico(true);
    }
    else
    {
        this.smsx.magnetico(false);
    }
}
else
{
    smsx.todo(false);
    System.out.println("Monitoreo SMS desactivado");
}
}

```

#### 5.3.3.3.9. Notificaciones Correo

En la siguiente figura, se muestra la interfaz de Notificaciones por Correo, la cual permite seleccionar 3 correos electrónicos distintos como destinatarios y configurar bajo qué eventos se envían estas



notificaciones, está restringida solo para el uso del usuario de tipo “Administrador”.

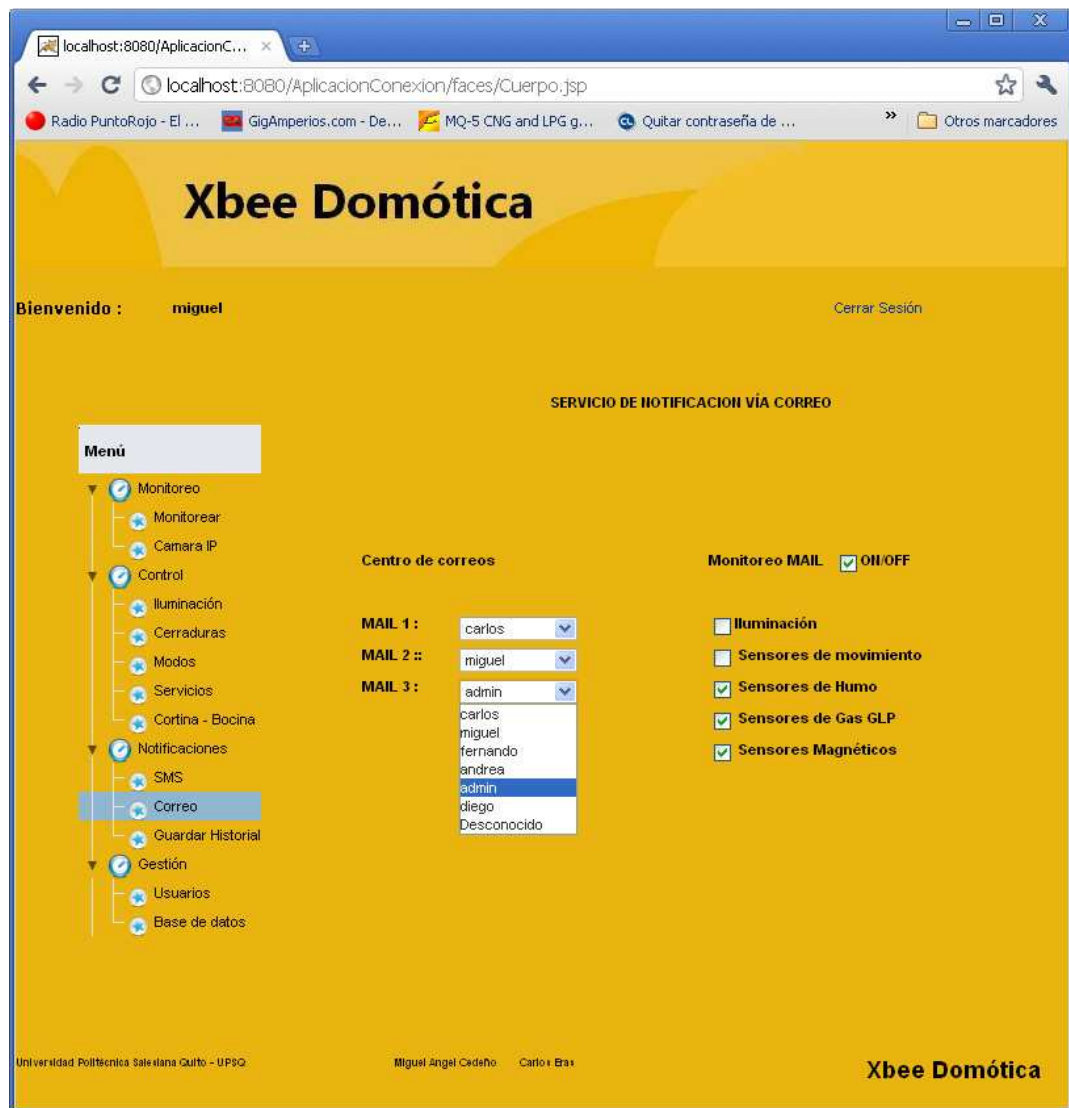


Figura 193: Interfaz Notificaciones Correo

En la parte de centro de correos existen 3 Combobox que permiten elegir usuarios de la base de datos, a los cuales les va a llegar las notificaciones a sus respectivos correos. En el siguiente fragmento de código, se muestra como se asignan estas direcciones al módulo servidor Zigbee y por medio de este se

comunica con el módulo servidor de correo para enviarlo de manera satisfactoria.

```
public void drop_mail1_processValueChange(ValueChangeEvent vce){
    xbt.asignar_mail(drop_mail1.getValue().toString(),
drop_mail2.getValue().toString(), drop_mail3.getValue().toString());
}
public void drop_mail2_processValueChange(ValueChangeEvent event){
    xbt.asignar_mail(drop_mail1.getValue().toString(),
drop_mail2.getValue().toString(), drop_mail3.getValue().toString());
}
public void drop_mail3_processValueChange(ValueChangeEvent event){
    xbt.asignar_mail(drop_mail1.getValue().toString(),
drop_mail2.getValue().toString(), drop_mail3.getValue().toString());
}
```

El monitoreo mail consiste en identificar bajo qué eventos se enviaran las notificaciones, para eso se crean checkbox con el fin de que el usuario seleccione estos eventos, en el siguiente fragmento de código, se muestra como están programados estos checkbox y como se comunican con el módulo servidor Zigbee.

```
public void monitoreomail(){
    if (chkMonitoreoMAIL.isChecked()){
        xbt.mail_todo(true);
        System.out.println("Monitoreo MAIL activado");
        //Monitoreo iluminacion
        if(chk_iluminacionMAIL.isChecked()){
            this.xbt.mail_iluminacion(true);
        }
        else
        {
            this.xbt.mail_iluminacion(false);
        }
        //Monitoreo movimiento
        if(chk_movMAIL.isChecked()){
            this.xbt.mail_mov(true);
        }
        else
```

```

{
    this.xbt.mail_mov(false);
}

```

#### 5.3.3.3.10. Notificaciones Guardar Historial

En la siguiente figura, se muestra la interfaz de Notificaciones Guardar Historial, la cual permite seleccionar bajo qué eventos se insertan datos en la tabla historial de la base de datos, está restringida solo para el uso del usuario de tipo “Administrador”.

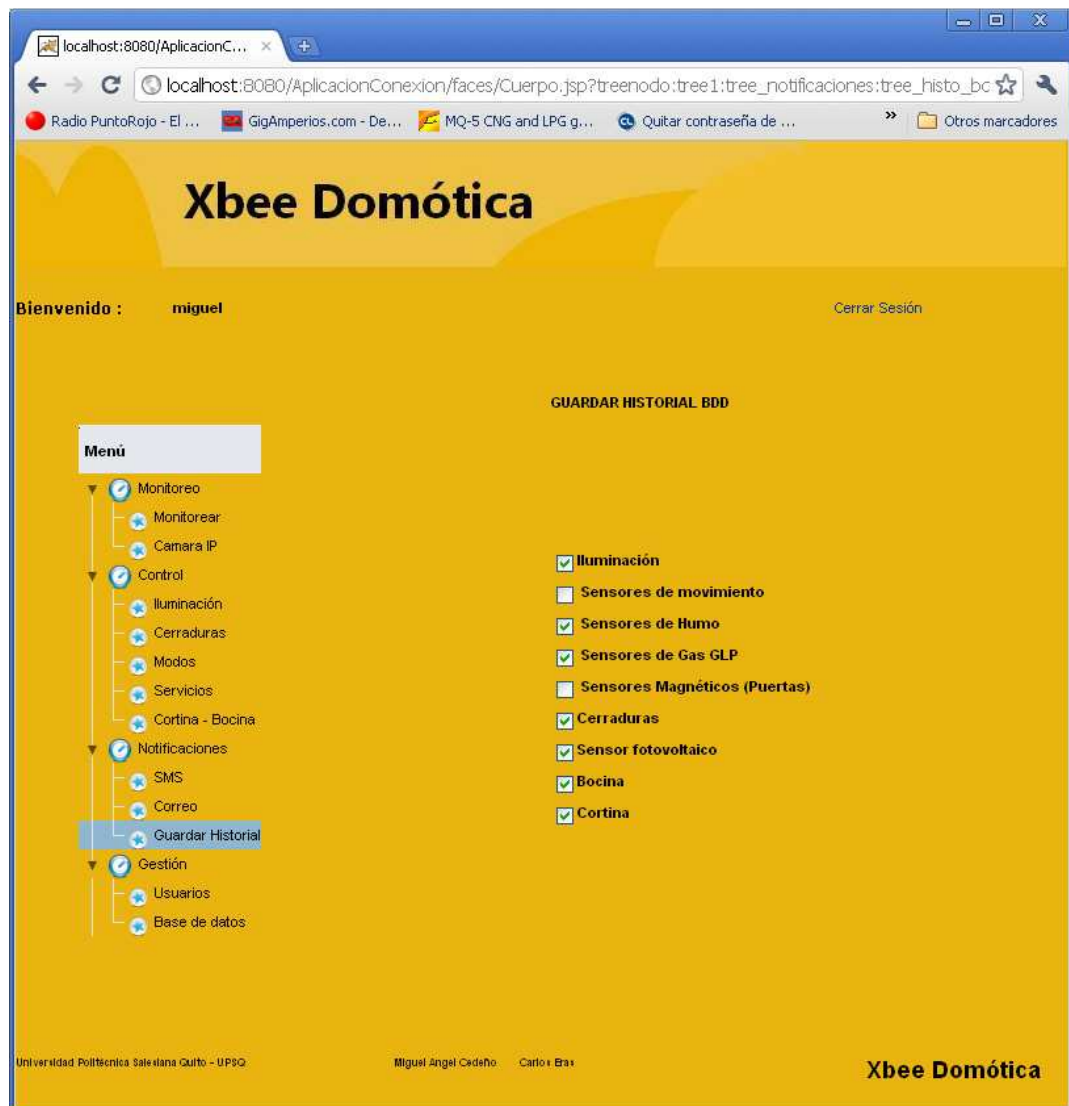


Figura 194: Interfaz Notificaciones Guardar Historial

Cuenta con 9 Checkbox los cuales identifican bajo que eventos se guarda el historial de la base de datos por ejemplo si se selecciona iluminación, esto quiere decir que si un foco sufre un cambio de estado, esta información se guardara en el historial de la base de datos.

En el siguiente fragmento de código, se muestra cómo se maneja la selección de estos checkbox y como se envía al módulo servidor Zigbee para que este los interprete y realice las acciones específicas.

```
public void setearmodos_grabar(){  
xbt.estado_grabar(chk_iluminacion.isChecked(),chk_mov.isChecked(),chk_hum  
o.isChecked(),chk_gas.isChecked(),chk_mag.isChecked(),chk_cerraduras.isCh  
echecked(),chk_fotovoltaico.isChecked(),chk_bocina.isChecked(),chk_cortina.isCh  
echecked());  
}
```

#### **5.3.3.3.11. Gestión Usuarios**

En esta interfaz se mostrará, se manipulará la tabla Usuarios de la base de datos mediante el servidor web, se podrá Insertar, Modificar, Eliminar y Consultar todos los usuarios que tienen de alguna forma acceso al sistema.

La interfaz Gestión de Usuarios está limitada al tipo de usuarios administradores solo ellos serán capaces de Insertar, Modificar, Eliminar y Consultar usuarios que estén almacenados en la base de datos.

#### **Consultar Usuarios**

Para la consulta de usuarios se utilizo Ajax, para facilitar el manejo de consultas.

A continuación, se muestra como se realizan las consultas mediante el uso de Ajax.

## Ajax con base de datos

1.- Se agrega los siguientes componentes un TextField, ListBox y un Button.

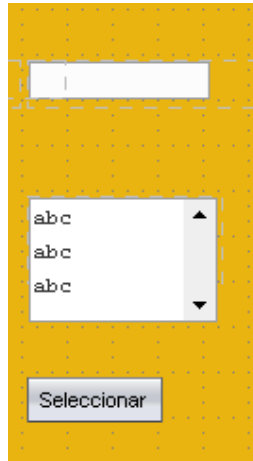


Figura 195: Consultar Usuario por medio de Ajax Paso 1

2.- Agregar la librería de componentes “Dynamic Faces” a su Entorno de Desarrollo (IDE).

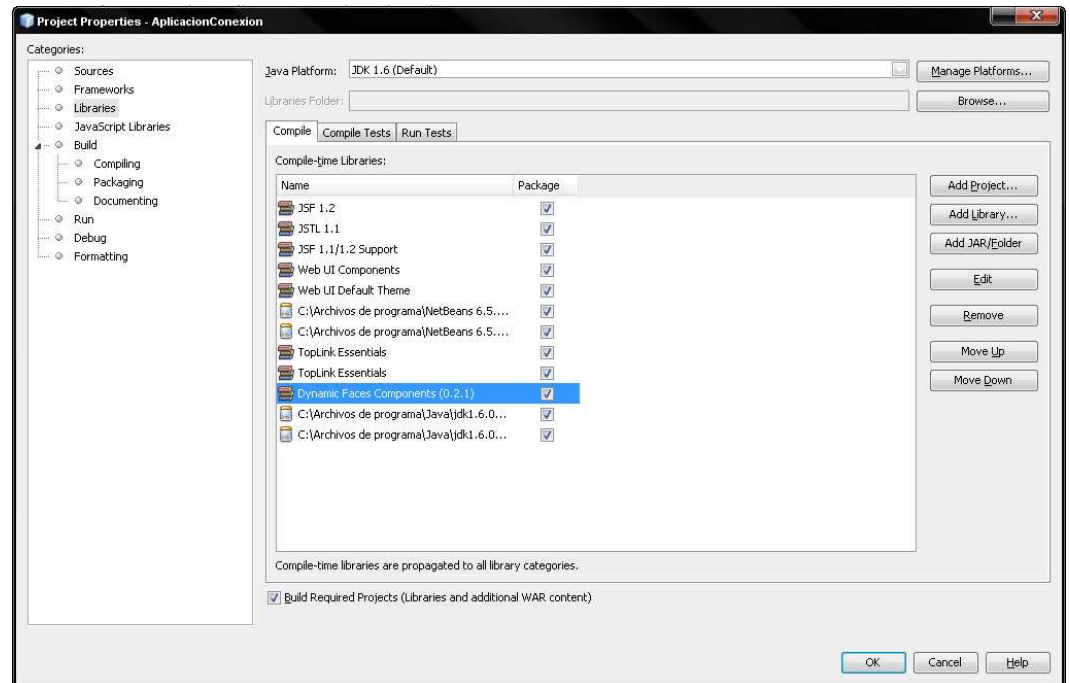


Figura 196: Consultar Usuario por medio de Ajax Paso 2

3.- Se agrega en la paleta de librería de componentes el nuevo elemento.



Figura 197: Consultar Usuario por medio de Ajax Paso 3

4.- Se agrega 3 Ajax Transaction al form\_usuarios.

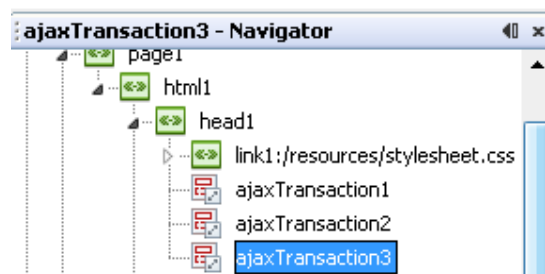


Figura 198: Consultar Usuario por medio de Ajax Paso 4

5.- Se configura Configura cada componente agregado.

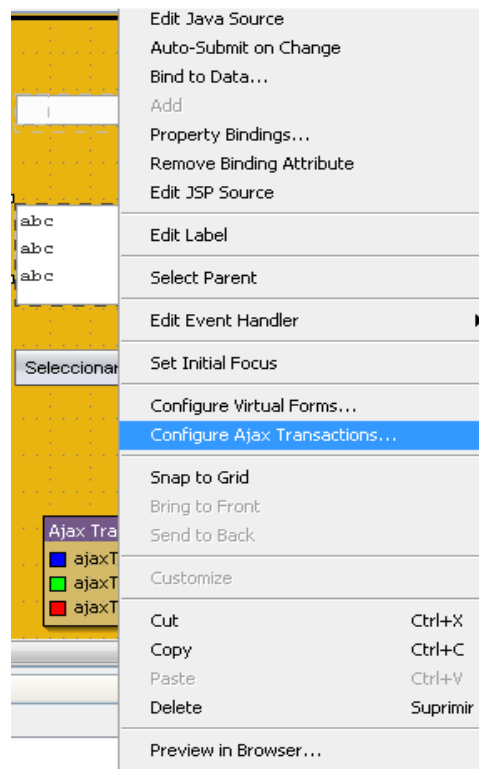


Figura 199: Consultar Usuario por medio de Ajax Paso 5

6.- Configurar el Ajax Transactions de cada componente.

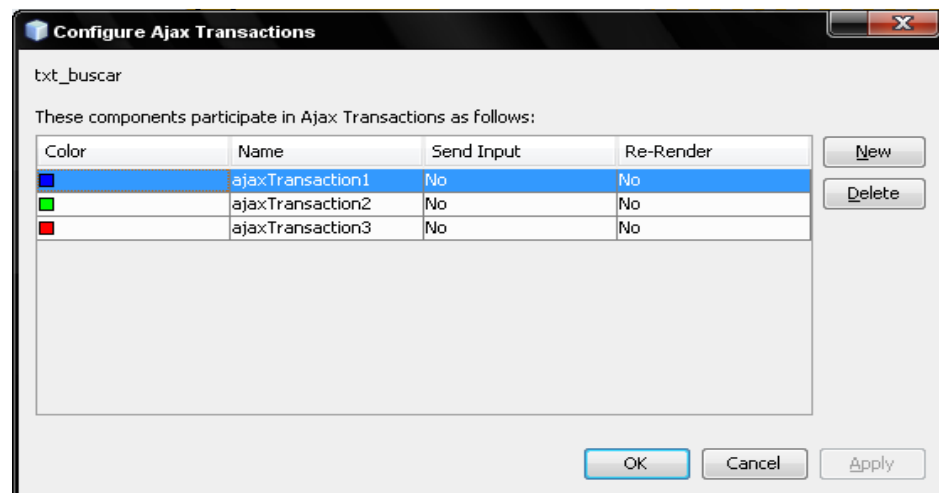


Figura 200: Consultar Usuario por medio de Ajax Paso 6

El componente “Ajax Transaction” permite configurar visualmente la funcionalidad Ajax, mostrando los diferentes componentes con un borde de colores ya sea sólido o punteado, esto dependerá si es una entrada que va a ser enviada al servidor será un borde sólido y cuando el cliente reciba una respuesta mediante Ajax el componente tendrá un borde punteado.

En el siguiente caso el Textfield será el que envíe la entrada al servidor y el Listbox conjuntamente con el Button serán mostrados cuando se reciba la respuesta mediante Ajax.

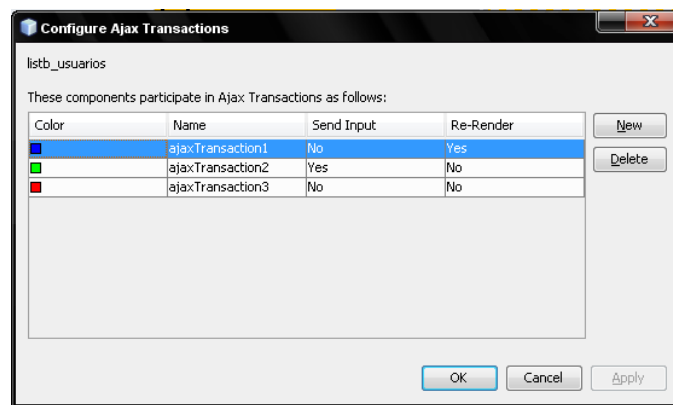


Figura 201: Configuración de los componentes mediante Ajax

7.- Se debe mostrar la siguiente estructura.

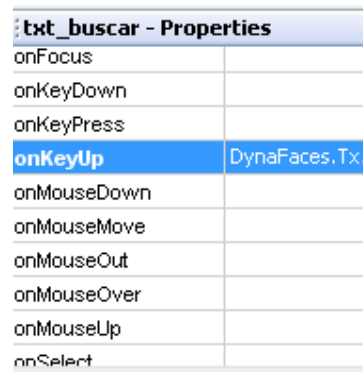


Figura 202: Consultar Usuario por medio de Ajax Paso 7



8.- Editar las propiedades de los compontes TextField y Listbox como se muestra en las siguientes figuras:

En el TextField



En el Listbox

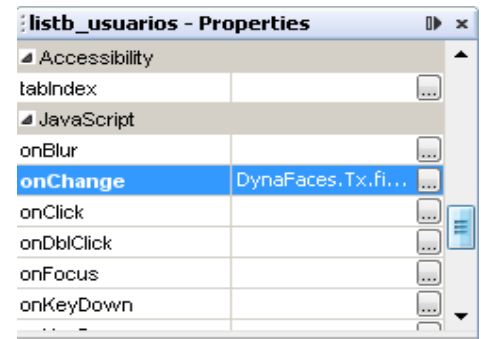


Figura 203: Consultar Usuario por medio de Ajax Paso 8

```
DynaFaces.Tx.fire("ajaxTransaction1", "textField1")
DynaFaces.Tx.fire("ajaxTransaction2", "listbox1")
```

## Interacción con la base de datos

Primero conectarse con la Base de Datos.

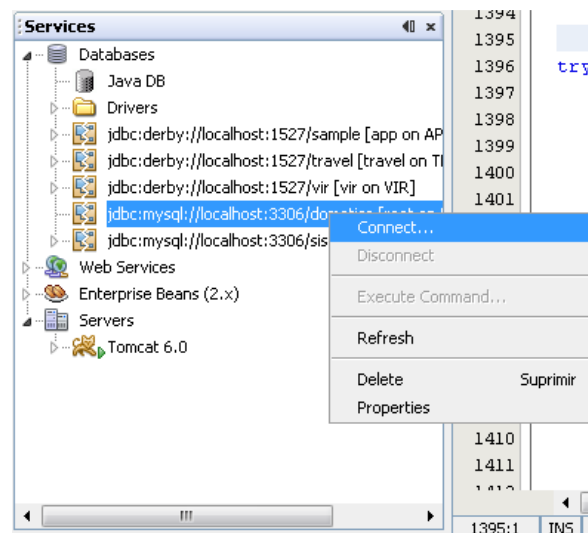


Figura 204: Conexión con la base de datos

Arrastrar la tabla usuarios (en éste caso) hacia el form\_usuarios en 2 ocasiones.

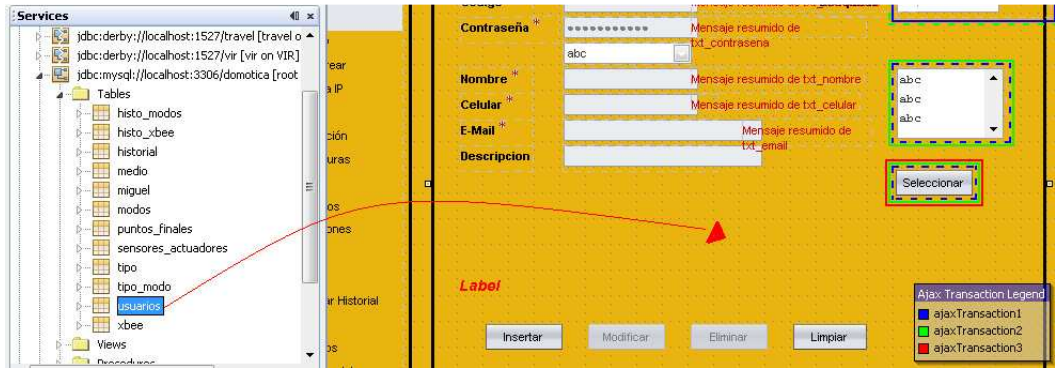


Figura 205: Insertar tabla en la interfaz Usuarios

Editar la consulta SQL la cual realizará la búsqueda que se requiere.

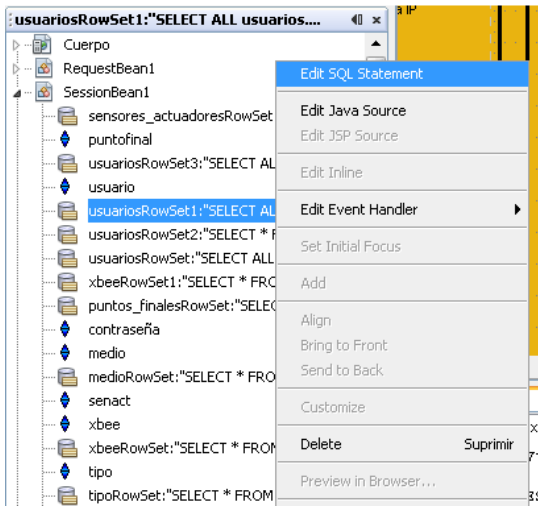


Figura 206: Editar sentencia SQL

Mediante la siguiente consulta se genera la búsqueda de usuarios.

```
SELECT ALL usuarios.`COD_USUARIO`, usuarios.`COD_TIPO`, usuarios.`NOMBRE`, usuarios.`DESCRIPCION` FROM usuarios WHERE usuarios.`COD_USUARIO` = ?
```

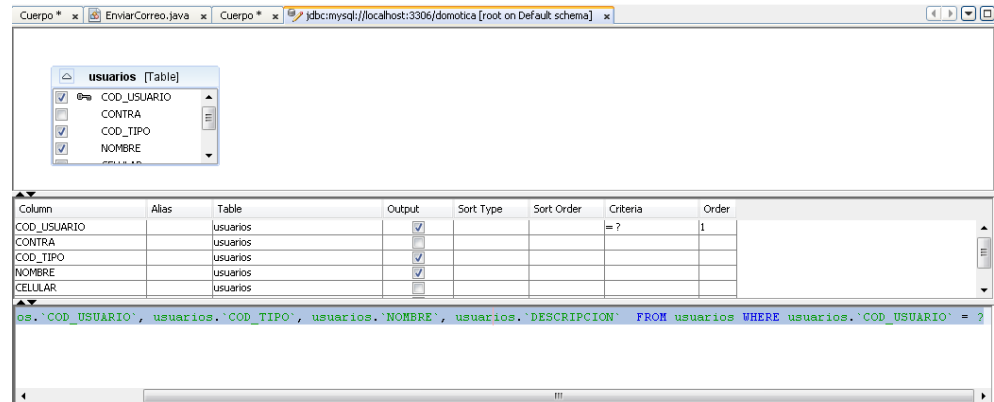


Figura 207: Interfaz donde se almacena la consulta SQL

Nuevamente editar el siguiente usuariosRowSet creado, pero ahora con la siguiente consulta:

```
SELECT ALL usuarios.`NOMBRE`, usuarios.`COD_USUARIO`, usuarios.`COD_TIPO`, usuarios.`DESCRIPCION` FROM usuarios WHERE usuarios.`NOMBRE` LIKE ?
```

Finalmente la acción que realizaran los componentes se muestra en el siguiente fragmento de código:

```
public void txt_buscar_processValueChanged(ValueChangeEvent event)
{
    if (txt_buscar.getText().toString() != null ||
    txt_buscar.getText().toString().length() > 0) {
        listb_usuarios.setVisible(true);
        btn_seleccionar.setVisible(true);
    }

    String prefix = txt_buscar.getText().toString();
    String doctName = prefix.substring(0, 1).toUpperCase() +
    prefix.substring(1) + '%';
    try {
        getSessionBean1().getUsuariosRowSet().setObject(1, doctName);
        usuariosDataProvider1.refresh();
    }
}
```

```

    } catch (SQLException ex) {
        Logger.getLogger(Cuerpo.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}
Código que se debe agregar al componente ListBox
public void listb_usuarios_processValueChanged(ValueChangeEvent
event) {
    if (txt_buscar.getText() == null ||
txt_buscar.getText().toString().length() == 0) {

        listb_usuarios.setVisible(false);
        btn_seleccionar.setVisible(false);
        try {
            getSessionBean1().getUsuariosRowSet().setObject(1, "");
        } catch (SQLException ex) {
            Logger.getLogger(Cuerpo.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
}
}

```

Nota: Se generará un error en `Logger.getLogger` porque requiere nuevas librerías, para solucionarlo ubicarse con el mouse en esta línea de código y presionar el botón secundario del mouse y escoger la opción `Fix Imports`, tal como se muestra en la siguiente figura.

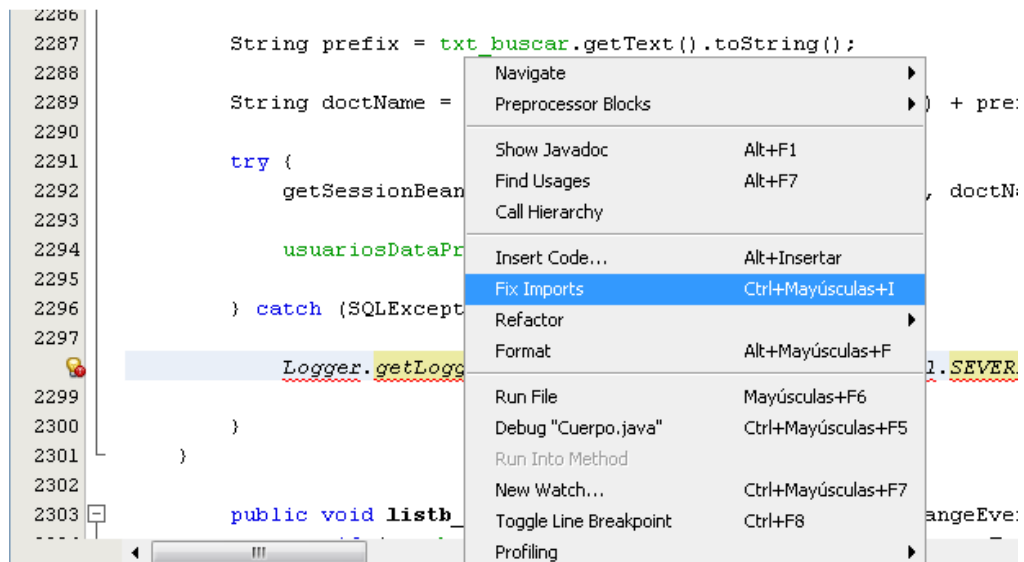


Figura 208: Agregar librerías mediante el `Fix Imports`

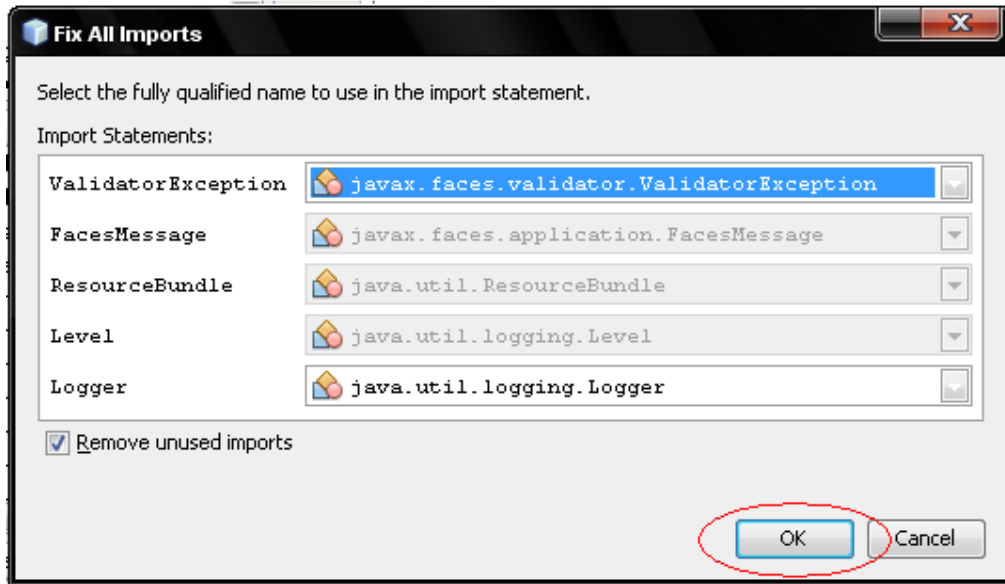


Figura 209: Agregar librerías Logger

Como último punto buscar el método “Prerender” e insertar el siguiente código:

```
if (txt_buscar.getText() == null || txt_buscar.getText().toString().length() == 0) {  
    listb_usuarios.setVisible(false);  
    btn_seleccionar.setVisible(false);  
    try {  
        getSessionBean1().getUsuariosRowSet().setObject(1, "");  
    } catch (SQLException ex) {  
        Logger.getLogger(Cuerpo.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

El Resultado Final es el que se muestra en la siguiente figura.

Busqueda

Figura 210: Agregar librerías Logger

La interfaz que será mostrada en el servidor web será la siguiente.

**Usuarios**

**Codigo**

**Usuario**

**Contraseña**

**Tipo**

**Celular**

**E-Mail**

**Descripcion**

**Busqueda**

**Consulta de usuario mediante Ajax**

Figura 211: Interfaz final de la consulta de usuario

## Insertar Usuarios

A continuación, se muestra cómo añadir un usuario al sistema para que tenga acceso al sitio web.

Figura 212: Interfaz Ingreso de Usuarios

Cuando se ingresa a la interfaz usuarios, todas las cajas de texto que se muestran están bloqueados para evitar insertar usuarios de forma equivocada.

Solo con presionar sobre el botón Insertar se activan las cajas de texto para insertar un nuevo usuario.

Los cuadro de texto Código, Usuario, e-mail, contraseña y celular están validados como campos obligatorios o requeridos, además se realizó validaciones numéricas, alfanuméricas y de texto, en las cajas de texto que presentan dificultades, a continuación se muestran las entradas admitidas para cada caja de texto.

Código: Únicamente 4 valores numéricos.

Celular: Solo caracteres de tipo numérico que comiencen en 09 – 08 y siete números a continuación de ellos, por ejemplo: 09xxxxxxx.

E-mail: Caracteres de texto seguidos del símbolo @ y al final separado con un punto, ejemplo: xxxxxx@xxxx.xxx.

Previamente ya realizando estas validaciones se puede acceder al botón Guardar caso contrario al observar campos erróneos éste botón no permitirá la inserción del nuevo usuario.

Este fragmento de código, muestra como se realiza la inserción de usuarios.

```
cxn.Ejecutar("INSERT INTO usuarios  
(COD_USUARIO,CONTRA,COD_TIPO,NOMBRE,CELULAR,MAIL,DESCRIPC  
ION) VALUES  
('"+txt_codusuario.getText()+"','"+txt_contrasena.getText()+"','"+cbx_tipo.getSel  
ected()+"','"+txt_nombre.getText()+"','"+txt_celular.getText()+"','"+txt_email.getT  
ext()+"','"+txt_descripcion.getText()+"");
```

### Eliminar usuarios

Para realizar la eliminación de un usuario previamente se debe consultar cual va a ser eliminado y cargar en las cajas de texto la información por medio del botón Seleccionar. Una vez cargada la información se presiona el botón Eliminar el cual generará la opción de Borrar o Cancelar la operación.

**Usuarios**

**Codigo** 4321

**Usuario** diego

**Contraseña** diego

**Tipo** Administrador

**Celular** 097015405

**E-Mail** tki\_cool@hotmail.com

**Descripcion** diego bob

**Busqueda**

diego  
Desconocido

Borrar

Cancelar



Figura 213: Interfaz eliminar Usuario

En el siguiente fragmento de código, se muestra como se realiza la eliminación de usuarios.

```
cxn.Ejecutar("DELETE FROM usuarios WHERE nombre="
+txt_nombre.getText()+ "");
```

### Modificar Usuarios

En ésta interfaz se escoge el campo o los campos que se desean modificar por medio de la activación de un Checkbox ubicado al lado izquierdo de cada caja de texto, por lo tanto si no se activa el Checkbox no se podrá modificar la información y las cajas de texto no serán activadas para ingresar texto.

Una vez realizado los cambios necesarios se generan dos opciones Actualizar o Cancelar la operación, quedando a criterio del usuario salir o efectuar cambios.

En la siguiente figura, se muestra la interfaz Modificar Usuarios.

**1**

**Usuarios**

☐ Codigo 4321

☒ Usuario \* diego

☒ Contraseña diego

☒ Tipo Visitante

☒ Celular \* 097015401

☒ E-Mail \* tki\_cool@hotmail.com

☒ Descripcion diego bobs

**Busqueda**

d diego Desconocido

**2**

Actualizar

Cancelar

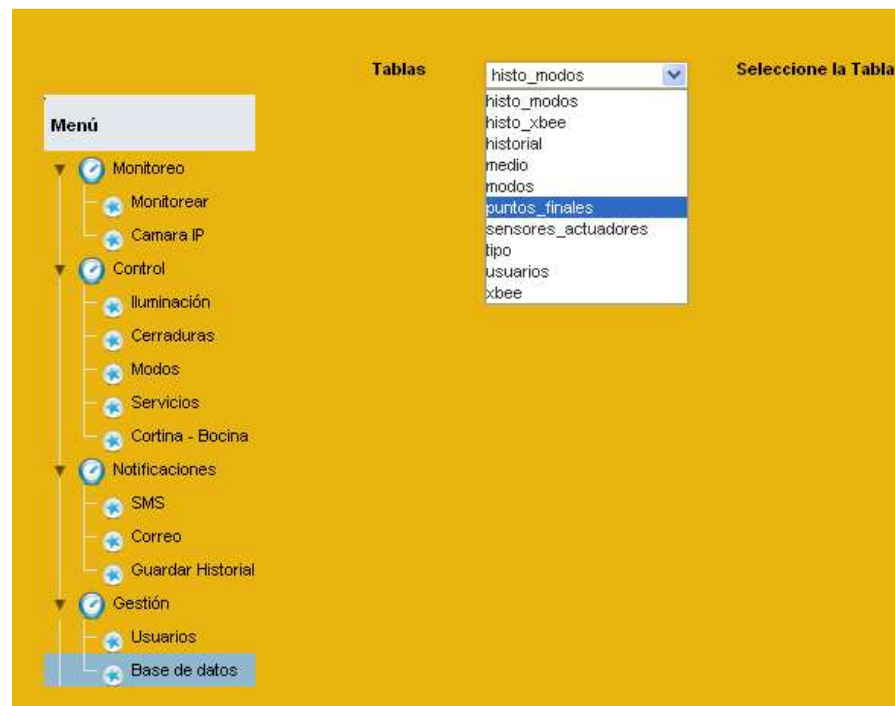
Figura 214: Interfaz eliminar Usuario

En el siguiente fragmento de código, se muestra como se realiza la eliminación de usuarios.

```
cxn.Ejecutar("UPDATE usuarios SET cod_usuario=" + txt_codusuario.getText()
+ ", contra=" + txt_contrasenalibre.getText()+ ",
cod_tipo=" + cbx_tipo.getSelected()+", nombre=" + txt_nombre.getText()+",
celular=" + txt_celular.getText()+", mail=" + txt_email.getText()+",
descripcion=" + txt_descripcion.getText()+ " WHERE
cod_usuario=" + lbl_codigoauxiliar.getText()+""");
```

#### 5.3.3.3.12. Gestión Base de datos

En esta interfaz se manipula las demás tablas que contiene la base de datos, así como también se realiza consultas personalizadas de datos específicos que se quiera conocer, en especial de la tabla historial que es donde se registran todos los eventos que realiza el sistema.



**Figura 215: Interfaz Gestión Base de Datos**

En esta interfaz por medio de un combo box se escoge la tabla que se requiere mostrar los datos de forma automática. Para mostrar todas las tablas de la base de datos se realiza la siguiente consulta SQL

```
SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE  
TABLE_TYPE = 'BASE TABLE' AND TABLE_SCHEMA = 'DOMOTICA'
```

### **Tabla histo\_modos**

Las consultas de esta tabla se realizan mediante 3 parámetros, por Fecha, por Modo y por usuario.

La consulta 1 se realiza mediante el componente calendario, existe para fecha de inicio y fecha final, esta consulta se genera al presionar en el botón Consultar.

Para la consulta 2 y 3 solo es requerido seleccionar el modo o el usuario y automáticamente se generará la consulta.

En todas las consultas se muestra el componente Table.

Para eliminar registros del historial basta con seleccionar uno, varios o todos los registros generados en la consulta, mediante los Checkbox que están dentro del componente Table.

Tablas histo\_modos Seleccione la Tabla

Fecha   
1

2
Modo MSP 3
Usuario carlos

**Historial Modos (1)**

☒ ☐

Codigo	Usuario	Estado	Modo	Fecha	Hora	
2	carlos	DESACTIVADO	MAE	10-nov-2010	12:02:24	<input type="checkbox"/>

Figura 216: Interfaz Tabla histo\_modos

En el siguiente fragmento de código, se muestra como se realizan las consultas y eliminación de registros de ésta tabla.

```
consultahmodos="SELECT v FROM Vistahistorialmodos v WHERE
v.modo='"+cbmodos+"'";
cxn.Ejecutar("DELETE FROM histo_modos WHERE cod_histo_modo = " +
cod_hmodos + "');
```

### Tabla histo\_xbee

Las consultas de ésta tabla se realizan mediante dos parámetros 1 es por fecha, y 2 por nombre de Xbee, la eliminación de registros generados se realiza de la misma manera que en la tabla histo\_modos.



## Tabla historial

Siendo la tabla historial la más compleja de todas, debido a que en ella se mostrará toda la información de los eventos que ocurren en el sistema, existen 5 parámetros para consultar la información que se requiere, 1 por fecha, 2 por Xbee, 3 por Estado, 4 por Usuario y 5 por Medio,

Como se muestra en la Figura 218, cada parámetro de búsqueda contiene un Checkbox el cual servirá para activar o desactivar el modo de consulta, por ejemplo si se tiene activo el parámetro 1,3,5 la consulta que se generara evaluara los campos Fecha, Estado y Medio, al presionar el botón Consultar automáticamente se realizará la consulta con los parámetros activados previamente. Para eliminar uno, varios o todos los registros generados por las consultas se requiere seleccionar los Checkbox que están dentro del componente Table y presionar el botón Eliminar registro seleccionados.

**Tablas** historial **Seleccione la Tabla**

1 ☐ Fecha 2 ☐ Xbee 3 ☐ Estado 4 ☐ Usuario 5 ☐ Medio

xbee1 ACTIVADO carlos Web

**Historial (1 - 6 de 1511)**

Historial	Usuario	Ubicacion	Medio	Xbee	Sensor Actuador	Estado	Fecha	Hora	
55	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	INACTIVO	15-sep-2010	12:02:27	<input type="checkbox"/>
56	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	ACTIVO	15-sep-2010	12:02:29	<input type="checkbox"/>
57	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	INACTIVO	15-sep-2010	12:02:31	<input type="checkbox"/>
58	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	ACTIVO	15-sep-2010	12:02:33	<input type="checkbox"/>
59	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	INACTIVO	15-sep-2010	12:02:36	<input type="checkbox"/>
60	Desconocido	cocina	Físico	xbee2	Sensor de movimiento 2	ACTIVO	15-sep-2010	12:02:38	<input type="checkbox"/>

« « Página: 1 de 252 » »

Figura 218: Interfaz Tabla historial

En el siguiente fragmento de código, se muestra como se realizan las consultas y eliminación de registros de ésta tabla.

```
cfecha="v.fecha BETWEEN '" + historialdesde + "' AND '" + historialhasta + "' AND ";
cxbee="v.xbee='" + historialxbee + "' AND ";
cestado="v.estado='" + historialestado + "' AND ";
cusuario="v.usuario='" + historialusuario + "' AND ";
cmedio="v.medio='" + historialmedio + "' AND ";
consultahistotial="SELECT v FROM Vistahistotial v WHERE "+cfecha+" "+cxbee+"
"+cestado+" "+cusuario+" "+cmedio+" 1=1";

cxn.Ejecutar("DELETE FROM historial WHERE COD_HISTORIAL = '" + cod_histo
+ "'");
```

### **Tabla medio**

En ésta tabla al tener definido previamente los 5 medios por los cuales se gestiona el sistema no cabe la posibilidad de eliminar ni insertar nuevos medios, salvo el caso que se realice una nueva versión del sistema, por éste motivo solo se contempla la posibilidad de mostrarlos y modificarlos o actualizarlos. Al seleccionar cualquier registro del componente Table por medio del RadioButton y presionar sobre el botón actualizar, se desplegarán las cajas de texto que contienen estos campos, y de esta forma modificar el texto que contengan.

**Tablas** medio **Seleccione la Tabla**

**Medio**

Codigo	Nombre	Descripción	
1	Web	Servidor Web	<input checked="" type="radio"/>
2	Bluetooth	Dispositivos Bluetooth	<input type="radio"/>
3	Control Touch	Control Remoto Inalambrico	<input type="radio"/>
5	Físico	Cajas físicas de control local	<input type="radio"/>
4	SMS	Via SMS mensajes de texto celular	<input type="radio"/>

**Actualizar**

**Codigo** 1  
**Nombre** Web  
**Descripción** Servidor Web

**Actualizar**

Figura 219: Interfaz Tabla medio

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
Medio newmedio = new Medio();
newmedio.setCodMedio((String)txt_bddmediocodigo.getText());
newmedio.setNombre((String) txt_bddmedionombre.getText());
newmedio.setDescripcion((String) txt_bddmediodescripcion.getText());
userControllerx.updatemedio(newmedio);
EntityManager em = getEntityManager();
try {
    em.getTransaction().begin();
    Medio mediox = em.find(Medio.class, medio.getCodMedio());
    mediox.setNombre(medio.getNombre());
    mediox.setDescripcion(medio.getDescripcion());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}
```



## Tabla modos

Los registros que se muestran en esta tabla también son previamente definidos y una inserción de otro modo conllevaría a una nueva versión del sistema, como también la eliminación de uno de ellos causaría defectos en el funcionamiento normal del sistema, por este motivo solo es posible realizar modificaciones a los registros siguiendo la misma metodología que en la tabla anterior. Es decir, paso 1,2,3,4 tal como se muestra en la siguiente figura.

Tablas modos Seleccione la Tabla

**Modos (3)**

Codigo	Nombre	Descripción	
1	MSP	Modo simulador de presencia	<span style="border: 2px solid green; border-radius: 50%; padding: 2px;">+</span> <b>1</b>
2	MAE	Modo ahorro de energia	○
3	MA	Modo alarma	○

Actualizar **2**

**Codigo** 1 **3**

**Nombre** MSP

**Descripción** Modo simulador de pre

Actualizar **4**

Figura 220: Interfaz Tabla modos

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
Modos newmodos = new Modos();
    newmodos.setCodModo((Integer)txt_bddmodoscodigo.getText());
    newmodos.setNombre((String) txt_bddmodosnombre.getText());
    newmodos.setDescripcion((String) txt_bddmodosdescripcion.getText());
    userControllerx.updatemodos(newmodos);
```

```

EntityManager em = getEntityManager();

try {
    em.getTransaction().begin();
    Modos modos = em.find(Modos.class, modo.getCodModo());
    modos.setCodModo(modo.getCodModo());
    modos.setNombre(modo.getNombre());
    modos.setDescripcion(modo.getDescripcion());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}

```

### **Tabla puntos\_finales**

En ésta tabla se muestran los puntos finales necesarios para el funcionamiento del sistema mediante ellos se puede saber desde que Xbee, ubicación, actuador o sensor realizó determinada tarea. Al ser tan importantes estos registros y para evitar un mal funcionamiento del sistema solo es posible modificarlos mas no insertar ni eliminar alguno de ellos.

Los pasos a seguir para modificar los registros se muestran en la siguiente figura.

Tablas puntos\_finales Seleccione la Tabla

**Puntos Finales**

Codigo	Cod Xbee	Cod Sensor Actuator	Ubicacion	Detalles	
112	1	12	sala	foco de la sala	<span style="border: 1px solid green; border-radius: 50%; padding: 2px;">●</span> 1
121	1	21	sala	sensor de movimiento sala	○
171	1	71	exterior	sensor de luz natural día noche	○
131	1	31	sala	sensor de humo sala de la casa	○

⏪ ⏴
Página: 1 de 4
⏵ ⏩

Actualizar 2

**Codigo** 112

**Cod Sensor Actuator** 12

**Detalles** foco de la sala

**Cod Xbee** 1

**Ubicación** sala

Actualizar 4

3

Figura 221: Interfaz Tabla puntos\_finales

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
EntityManager em = getEntityManager();
try {
    em.getTransaction().begin();
    PuntosFinales puntofinall = em.find(PuntosFinales.class,
puntofinal.getCodPunto());
    puntofinall.setCodXbee(puntofinal.getCodXbee());
    puntofinall.setCodSenact(puntofinal.getCodSenact());
    puntofinall.setUbicacion(puntofinal.getUbicacion());
    puntofinall.setDetalles(puntofinal.getDetalles());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}
```

```

PuntosFinales newpfinales = new PuntosFinales();
newpfinales.setCodPunto((String) txt_bddpfinalescodigo.getText());
newpfinales.setCodXbee((String) txt_bddpfinalesxbee.getText());
newpfinales.setCodSenact((String) txt_bddpfinalesessenact.getText());
newpfinales.setUbicacion((String) txt_bddpfinalesubicacion.getText());
newpfinales.setDetalles((String) txt_bddpfinalesdetalles.getText());
userControllerx.updatepuntofinal(newpfinales);

```

## Tabla sensores\_actuadores

Los registros que en esta tabla se almacenan de igual forma son fijos y la inserción o eliminación de uno de ellos conllevaría a una mala respuesta por parte del servidor, por esta razón solo se permiten modificaciones.

Para realizar modificación de registros en esta tabla se siguen los pasos que se muestran en la siguiente figura.

The screenshot shows a web application interface for managing the 'sensores\_actuadores' table. The interface includes a table with columns for 'Codigo', 'Nombre', and 'Descripcion'. A red box highlights the table area, and a green box highlights the 'Actualizar' button. The table contains the following data:

Codigo	Nombre	Descripcion
12	Foco Sala	foco de la sala
21	Sensor de movimiento 1	detector de movimiento sala
71	Sensor fotovoltaico	sensor de luz natural
31	Sensor de humo	detecta humo

Below the table, there is a form for editing a record. The form includes fields for 'Codigo', 'Nombre', and 'Descripcion', and an 'Actualizar' button. The form is labeled with a green box and the number 3. The 'Actualizar' button is labeled with a green box and the number 4. The table is labeled with a green box and the number 1. The 'Actualizar' button at the top is labeled with a green box and the number 2.

Figura 222: Interfaz Tabla sensores\_actuadores

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
EntityManager em = getEntityManager();

try {
    em.getTransaction().begin();
    SensoresActuadores senactt = em.find(SensoresActuadores.class,
senact.getCodSenact());
    senactt.setNombre(senact.getNombre());
    senactt.setDescripcion(senact.getDescripcion());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}

SensoresActuadores newsen = new SensoresActuadores();
newsen.setCodSenact((String) txt_bddsenactcodigo.getText());
newsen.setNombre((String) txt_bddsenactnombre.getText());
newsen.setDescripcion((String)txt_bddsenactdetalles.getText());
userControllerx.updatesenact(newsen);
```

### **Tabla tipo**

Esta tabla cuenta con 2 registros Administrador y Visitante, es así como se define en el diseño de este sistema, las dos maneras que un usuario tendrá acceso al sistema, por este motivo solo se podrá modificar estos dos registros mas no insertar o peor eliminar uno de ellos.

Para modificar siga los pasos que se muestran en la siguiente figura.



Figura 223: Interfaz Tabla tipo

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
Tipo newtipo = new Tipo();
newtipo.setCodTipo((String) txt_bddtipocodigo.getText());
newtipo.setTipoUsuario((String) txt_bddtipotipousuario.getText());
newtipo.setDetalles((String) txt_bddtipodetalles.getText());
userControllerx.updatetipo(newtipo);
EntityManager em = getEntityManager();

try {
    em.getTransaction().begin();
    Tipo tipox = em.find(Tipo.class, tipo.getCodTipo());
    tipox.setTipoUsuario(tipo.getTipoUsuario());
    tipox.setDetalles(tipo.getDetalles());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}
```

## Tabla Usuarios

Para la consulta de ésta tabla se hace visible el nodo tree usuarios que se explica más ampliamente en el punto Gestión Usuarios.

## Tabla Xbee

El sistema contiene únicamente 4 módulos Xbee encargados de realizar diferentes tareas cada uno, es por esto que insertar uno nuevo o eliminar uno de ellos conllevaría a fallas graves en el sistema motivo por el cual solo es permitida la modificación de uno de ellos.

Al seguir los pasos que se muestran en la siguiente figura se realiza la modificación de los registros de la tabla Xbee.

**Tablas** xbee **Seleccione la Tabla**

**Xbee (4)**

Codigo Xbee	Nombre	Descripcion	
1	xbee1	Ubicado en la Sala controla : Foco Sala , Detector de movimiento 1, Sensor Fotovoltaico, Detector de humo, Bocina, Cortina	<input checked="" type="radio"/>
2	xbee2	Ubicado en la Cocina controla : Foco dormitorio 1 ,Foco cocina, Detector de mov 2,Detector de gas	<input type="radio"/>
3	xbee3	Ubicado en la Sala controla : Foco Exterior, Sensor Magnético Puerta 1, Sensor Magnético Puerta 2, Cerradura 1 Puerta 1,Cerradura 2 Puerta 2	<input type="radio"/>
4	xbee4	Control Inalámbrico Touch	<input type="radio"/>

**Actualizar**

**Codigo** 1  
**Nombre** xbee1  
**Descripcion** Ubicado en la Sala con

**Actualizar**

Figura 224: Interfaz Tabla xbee

En el siguiente fragmento de código, se muestra como se realiza la modificación de registros de ésta tabla.

```
Xbee newxbee = new Xbee();
newxbee.setCodXbee((String) txt_bddxbeeCodigo.getText());
newxbee.setNombre((String) txt_bddxbeenombre.getText());
newxbee.setDescripcion((String) txt_bddxbeedescripcion.getText());
userControllerx.updatexbee(newxbee);
EntityManager em = getEntityManager();
try {
    em.getTransaction().begin();
    Xbee xbeex = em.find(Xbee.class, xbee.getCodXbee());
    xbeex.setCodXbee(xbee.getCodXbee());
    xbeex.setNombre(xbee.getNombre());
    xbeex.setDescripcion(xbee.getDescripcion());
    em.getTransaction().commit();
} finally {
    em.close();
    return false;
}
```



#### 5.3.3.4. Seguridades

Las seguridades adoptadas para el servidor principal son:  
Encriptación y respaldos de la base de datos.

#### 5.3.3.5. Encriptación

La encriptación se realiza solo en la tabla usuarios, en el campo contraseña, de esta manera si un usuario desconocido logra entrar a la base de datos no podrá visualizarla.

En la siguiente figura, se muestra la tabla usuarios sin encriptación, es muy sencillo ver el campo contra y de esta manera obtener la contraseña de todos los usuarios.

+ Opciones

			COD_USUARIO ▾	NOMBRE	CONTRA	COD_TIPO	CELULAR	MAIL	DESCRIPCION
<input type="checkbox"/>			9999	Desconocido		1			Usuario desconocido
<input type="checkbox"/>			6666	admin	admin	1	097156156	mihu@h.com	Administrador
<input type="checkbox"/>			5678	andrea	123456	2	091778632	andreitamin@hotmail.com	Visitante FAMILIAR
<input type="checkbox"/>			4321	diego	diego	1	097015405	tiki_cool@hotmail.com	Hijo mayor

Figura 225: Tabla Usuarios sin encriptación

El siguiente fragmento de código, inserta un nuevo usuario en la base de datos con su contraseña encriptada, la clave de encriptación que se usa es 'abc123a'.

```
INSERT INTO usuarios VALUES(1112,'miguel', AES_ENCRYPT('angel','abc123a'),  
'1','097015405','miguelangel_cede@hotmail.com','administrador papa')
```

En la siguiente figura, se muestra la tabla usuarios con el campo contraseña encriptado.

		COD_USUARIO ▲	NOMBRE	CONTRA	COD_TIPO	CELULAR	MAIL	DESCRIPCION
<input type="checkbox"/>		1111	fernando	j)l+čN/U  Γ■	1	097589636	f_dopaz@hotmail.com	administrador tio
<input type="checkbox"/>		1112	miguel	č LÜLg)N=1Σ:	1	097015405	miguelangel_cede@hotmail.com	administrador papa
<input type="checkbox"/>		1113	carlos	Fë[Θ=ΣGf`g°@±	2	099895623	f_dopaz@hotmail.com	invitado vecino
<input type="checkbox"/>		1114	diego	lr(*s1²;4r1q	2	096895697	diego_nob@hotmail.com	invitado

Figura 226: Tabla Usuarios con encriptación

El siguiente fragmento de código, muestra como desencriptar los campos contraseña usando la clave de encriptación.

```
SELECT cod_usuario , nombre, AES_DECRYPT(contra,'abc123a') as
contra_desencriptada FROM usuarios
```

cod_usuario ▲	nombre	contra_desencriptada
1111	fernando	paz
1112	miguel	angel
1113	carlos	carlos
1114	diego	diego1234

Figura 227: Consulta a la tabla usuarios desencriptación

### 5.3.3.6. RespalDOS

Para generar un respaldo de la base de datos se usa el ejecutable mysqldump, desde java es llamado de la siguiente manera.

```
Process child = Runtime.getRuntime().exec("cmd /c mysqldump --opt --password= -
-user=root DBName > " + ubicacion);
```

Para respaldar la base de datos del sistema Zigbee domótica se ejecuta este comando.

```
c:\xampp\mysql\bin\mysqldump --opt --password=root123456 -u root domotica >
c:\respaldosbdd\respaldo.sql
```

Ingresa 4 parámetros básicos, la contraseña de MySQL, el nombre de usuario en este caso root, el nombre de la base de datos y la ruta con el nombre donde se va a guardar el respaldo.

## CAPÍTULO 6

### PRUEBAS

Entre las pruebas que se pueden realizar a un sistema se han seleccionado las siguientes.

#### 6.1. PRUEBAS DE COMPATIBILIDAD

Las pruebas de compatibilidad ayudan a determinar si el producto funcionará correctamente en ambientes distintos a los cuales fue desarrollado en un principio.

El servidor principal fue probado en un principio en el navegador Firefox 3.6.4, pero la idea es que funcione con otros navegadores, se probó el sistema en Google Chrome 7.0.517.44 y en internet Explorer obteniendo un resultado favorable sin ningún tipo de complicación.

También se realizó la prueba de compatibilidad en el navegador Opera Mobile 10, en un teléfono inteligente Nokia e63 mostrando muy buenos resultados, el único inconveniente que se presentó, fue el no permitir la visualización de la cámara IP.

En las siguientes figuras, se muestran las pruebas con el teléfono Nokia e63.

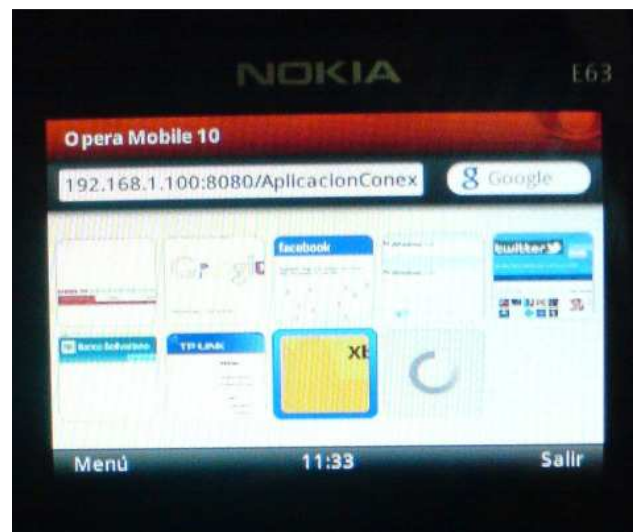


Figura 228: Prueba Compatibilidad Nokia e63 y Navegador Opera Mobile 10



Figura 229: Prueba Compatibilidad Nokia e63 y Navegador Opera Mobile 10 Ingreso al sistema



Figura 230: Prueba Compatibilidad Nokia e63 y Navegador Opera Mobile 10 Control de Iluminación

## **6.2. PRUEBAS DE SEGURIDAD**

La seguridad es un aspecto muy importante en el desarrollo de este sistema porque uno de los objetivos principales es hacer un hogar más seguro. Se llevaron a cabo diferentes pruebas para garantizar la seguridad del sistema.

Tales como la encriptación de la contraseña en la base de datos, Respaldos periódicos de la base de datos , control de usuarios en el servidor web, en el dispositivo GLCD táctil, Aplicación Bluetooth y SMS.

Se realizaron estas pruebas sin novedades para el sistema, se puso de manifiesto una confusión en la contraseña, que se debe ingresar en el módulo GLCD debido a que es posible solo ingresar datos numéricos para registrar al usuario, se debe ingresar el código de usuario no la contraseña, de esta manera se tiene acceso al control inalámbrico táctil.

## **6.3. PRUEBAS DE LA BASE DE DATOS**

Para realizar las pruebas de base de datos se tomó en cuenta todas las consultas que realiza el servidor web para obtener información y mostrarla según requiera el usuario, por medio de estas pruebas se corrigió errores existentes hasta llegar al punto que todas las consultas sean eficientes, claras y sobre todo funcionales al instante de mostrar la información.

**Nota:** La tabla de pruebas de base de datos se encuentra en el anexo 19.

## **6.4. PRUEBAS DE CONECTIVIDAD ZIGBEE**

Las pruebas de conectividad Zigbee permite tener una idea clara del alcance de estos módulos y de la capacidad de convergencia que poseen cuando están fuera del alcance del coordinador usan los Router para llegar a él.

## Pruebas de alcance con línea de vista sin obstáculos

Para la realización de estas pruebas se usa un bucle en el Xbee final es decir, se conecta entre si el pin de transmisión y de recepción, de esta manera los datos que el coordinador envía al cliente serán devueltos.

En las siguientes figuras, se muestra el test realizado a 1, 10, 50, 75, 85, 95 y 100 metros del coordinador.

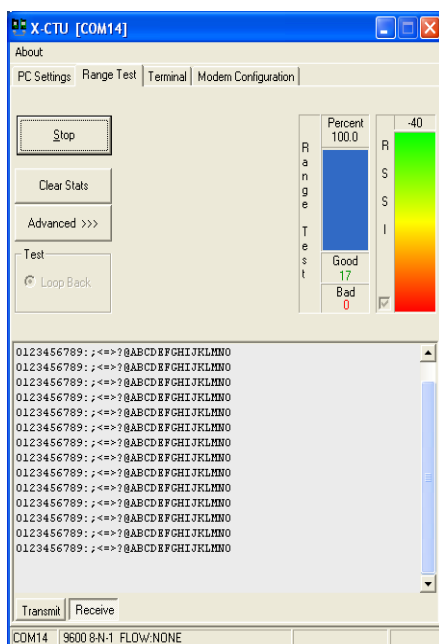


Figura 231: Test a 1 metro del coordinador

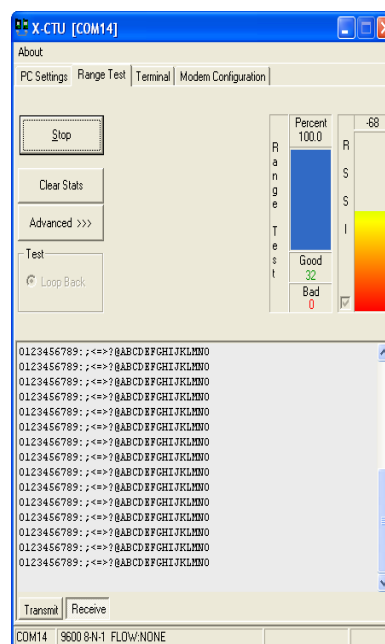


Figura 232: Test a 10 metros del coordinador

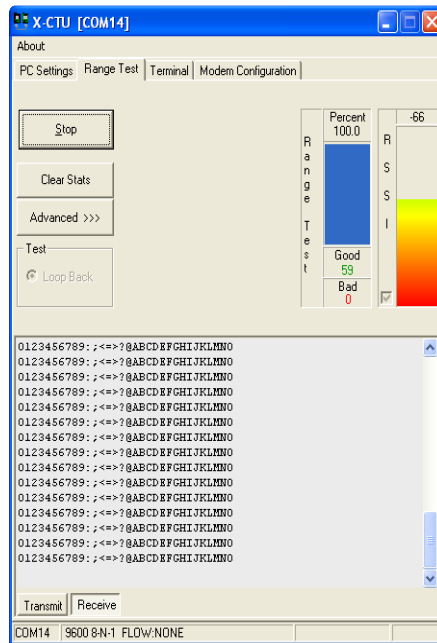


Figura 233: Test a 50 metros del coordinador

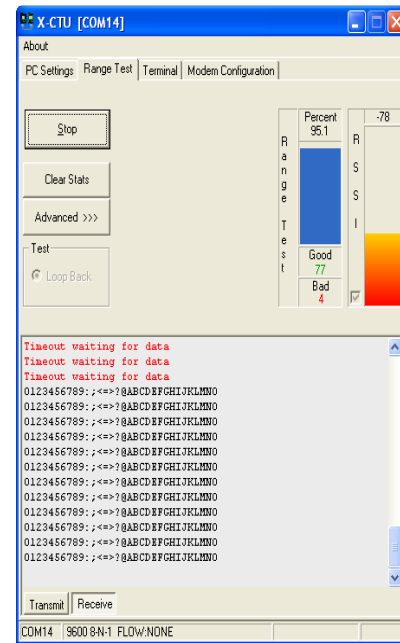


Figura 234: Test a 75 metros del coordinador

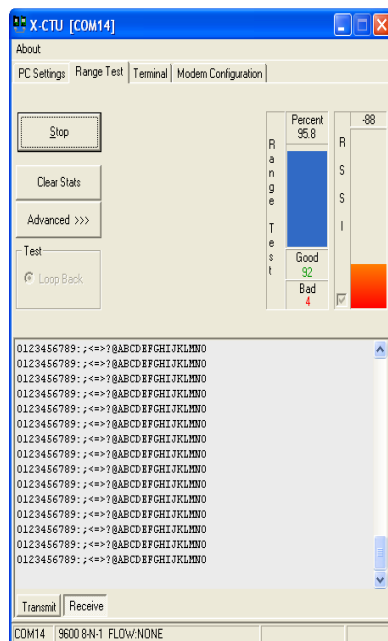


Figura 235: Test a 85 metros del coordinador

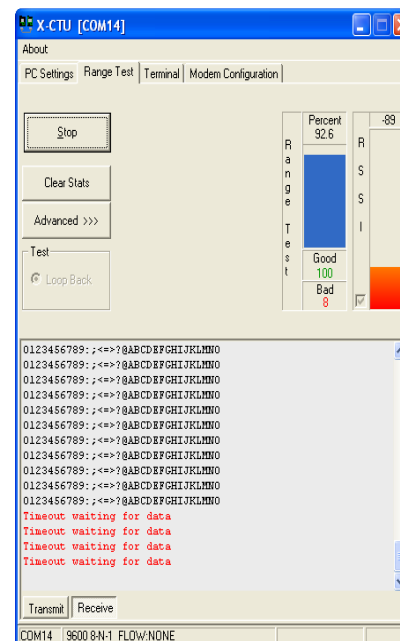
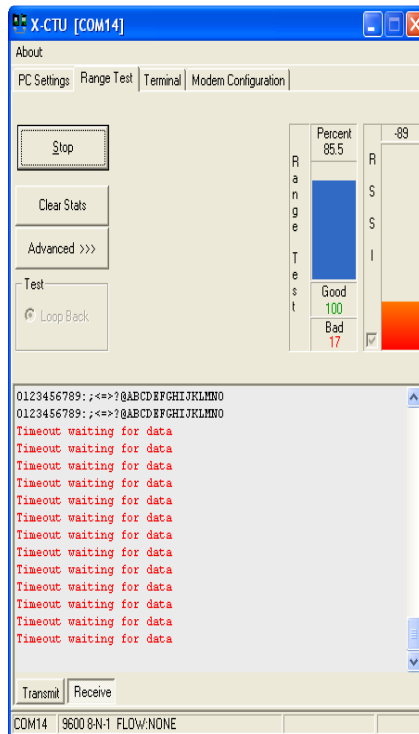


Figura 236: Test a 95 metros del coordinador



**Figura 237: Test a 100 metros del coordinador**

En esta prueba se concluyó que el alcance de un módulo Xbee serie 2 es de 100 metros pero lo ideal es hacerlo funcionar máximo hasta 90 metros con línea de vista sin obstáculos.

### **Pruebas de conectividad en la distribución física**

Se realizo una prueba de conectividad en la distribución física de cada módulo en el hogar y estos fueron los resultados.



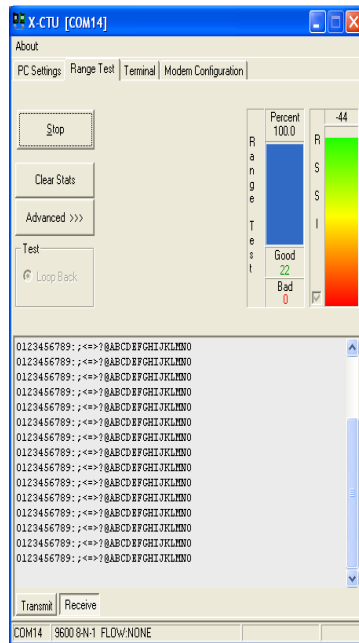


Figura 238: Test en el cuarto del servidor

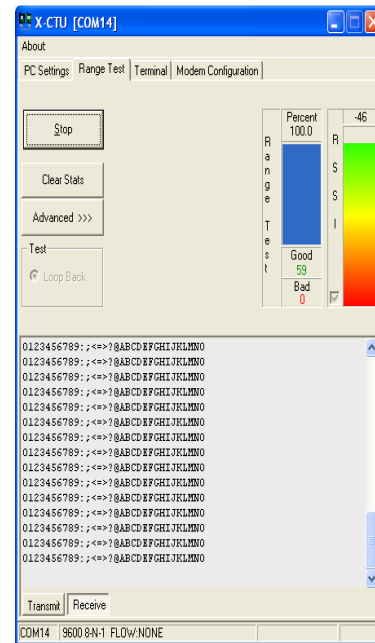


Figura 240: Test en la Sala

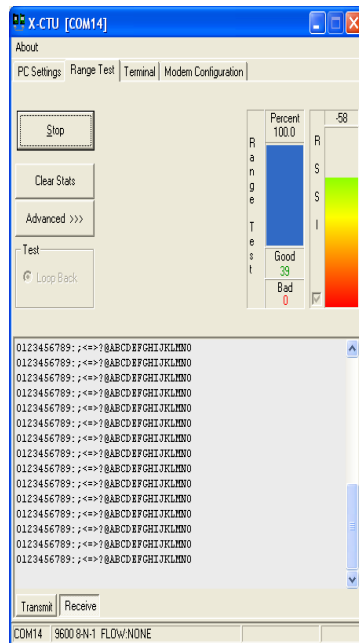


Figura 239: Test en el Dormitorio 1

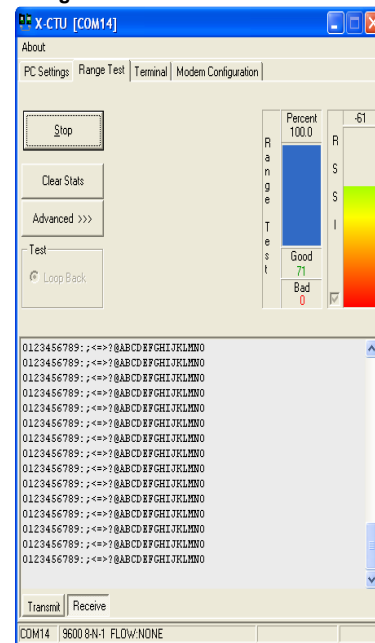


Figura 241: Test en la cocina

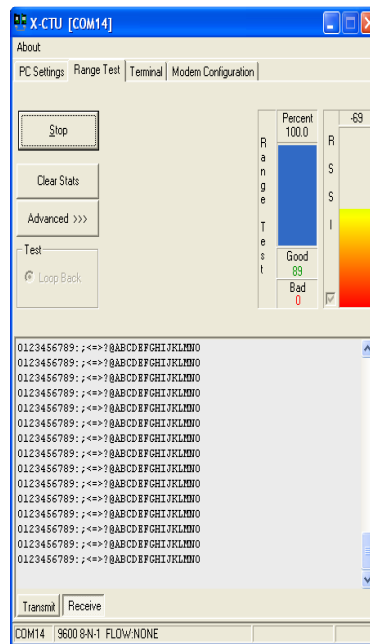


Figura 242: Test en el Exterior

## Prueba de la red Mesh

Si un dispositivo excede el alcance máximo de conectividad es recomendable colocar otro módulo Xbee series 2 en la mitad por ejemplo.

En la siguiente figura, se puede observar que el Xbee 6 se encuentra a 400 metros del Xbee 2 pero se podrá comunicarse por medio del Xbee 5, Xbee 4 y Coord. Estos módulos están configurados para ser Router son capaces de determinar a sus vecinos y converger cuando uno de ellos falle.

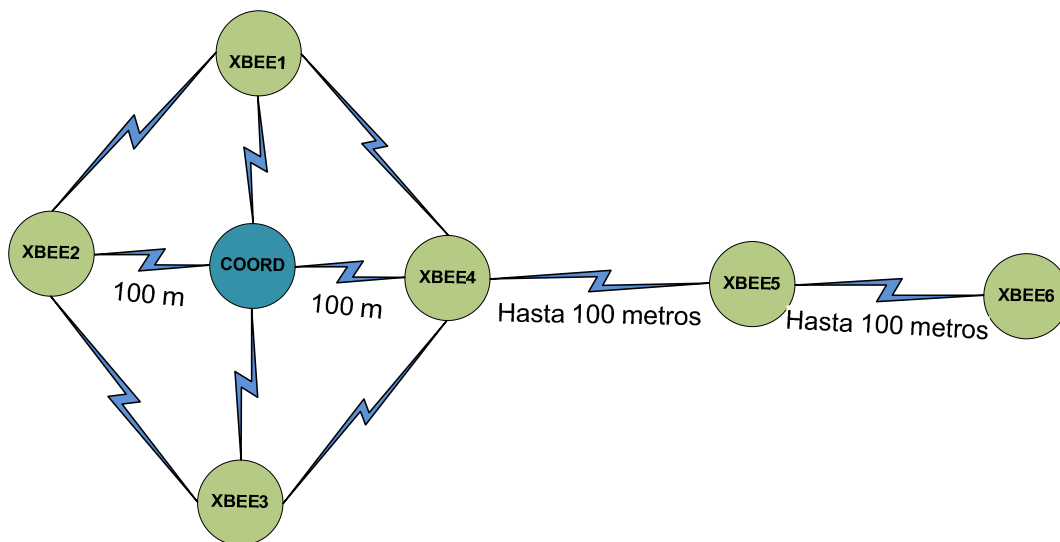


Figura 243: Prueba Red MESH

## 6.5. PRUEBAS DE FUNCIONAMIENTO

### 6.5.1. BLUETOOTH

Las pruebas del funcionamiento de la aplicación bluetooth fueron realizadas escuchando un Hyperterminal y respondiendo a sus peticiones por este medio.

En la siguiente figura, se puede observar en la primera línea que la aplicación bluetooth fue iniciada, se recibe HOL separado por puntos, Cuando el usuario presiona el botón emergencia llega la palabra “EMERGENCIA” al terminal abierto, al ingresar el usuario y la contraseña llega el usuario separado por un espacio y un punto de la contraseña, para validar se envía la ‘s’ si es correcto y ‘n’ si es incorrecto, se prueba presionando el botón “Monitorear” y la aplicación envía la palabra “ESTADOS” de esta manera realiza la petición de monitoreo, si se presiona cada uno de los botones de control de

iluminación la aplicación envía los códigos que encienden o apagan los distintos focos.

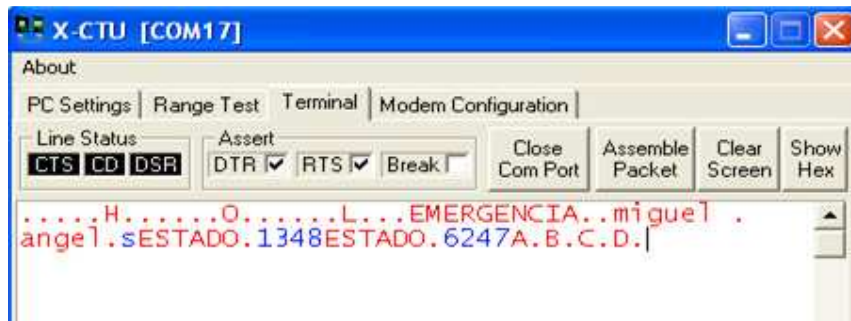


Figura 244: Hyperterminal Bluetooth

## 6.5.2. MÓDULO XBEE1

Las pruebas realizadas en este módulo se muestran en la siguiente figura, básicamente se verifica que el módulo envíe correctamente los códigos de comunicaciones.

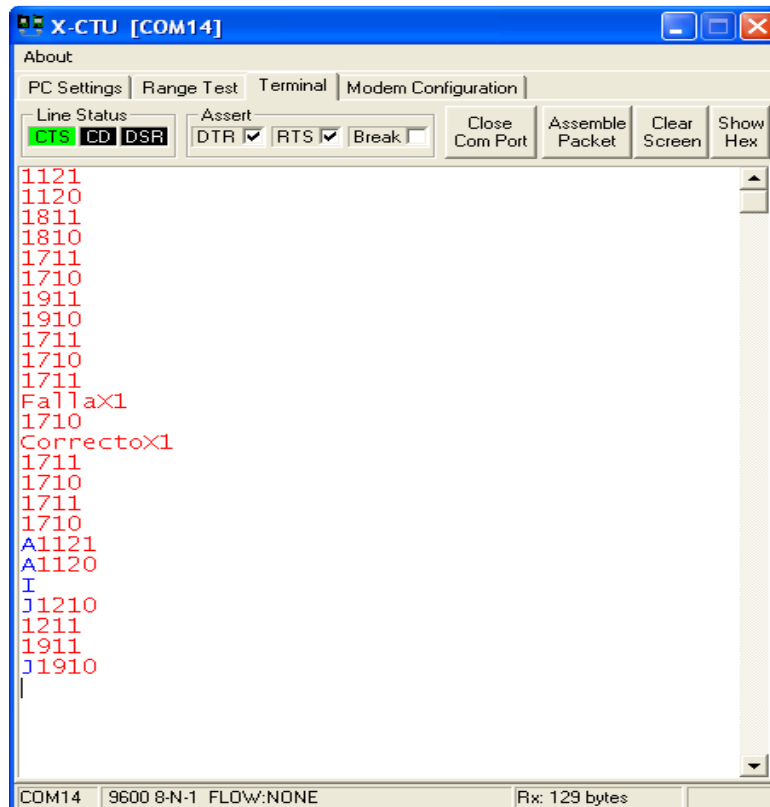


Figura 245: Pruebas funcionamiento Módulo Xbee 1

### 6.5.3. MÓDULO XBEE2

Las pruebas realizadas en este módulo se muestran en la siguiente figura, básicamente se verifica que el módulo envíe correctamente los códigos de comunicaciones.

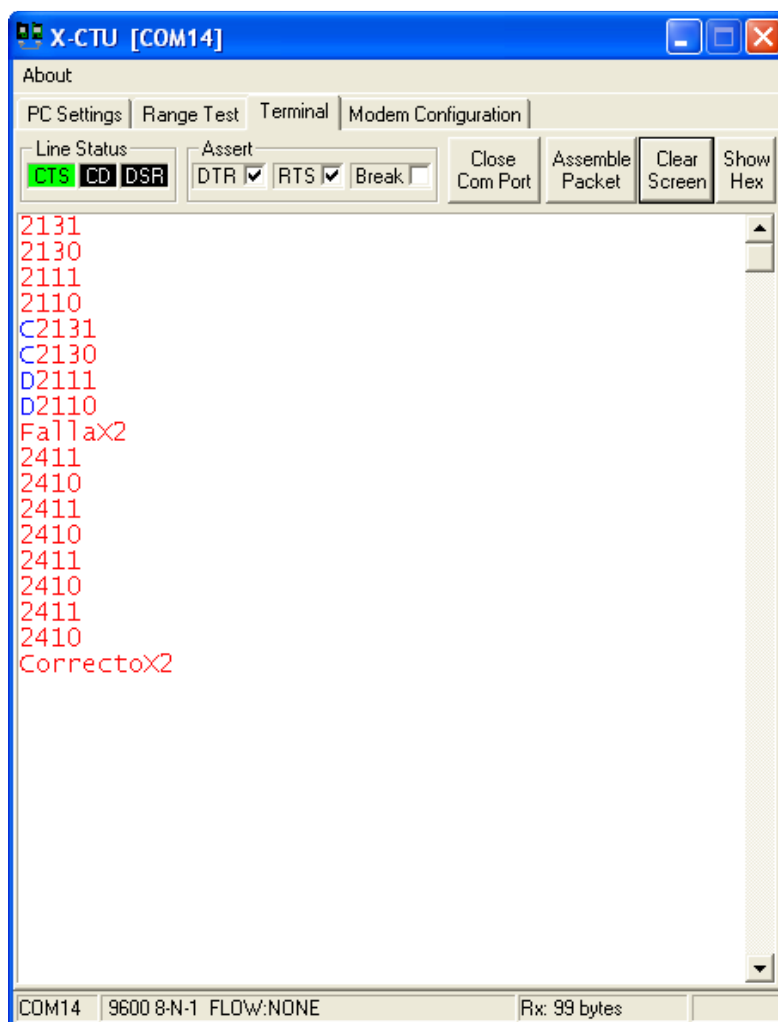


Figura 246: Pruebas funcionamiento Módulo Xbee 2

#### 6.5.4. MÓDULO XBEE3

Las pruebas realizadas en este módulo se muestran en la siguiente figura, básicamente se verifica que el módulo envíe correctamente los códigos de comunicaciones.

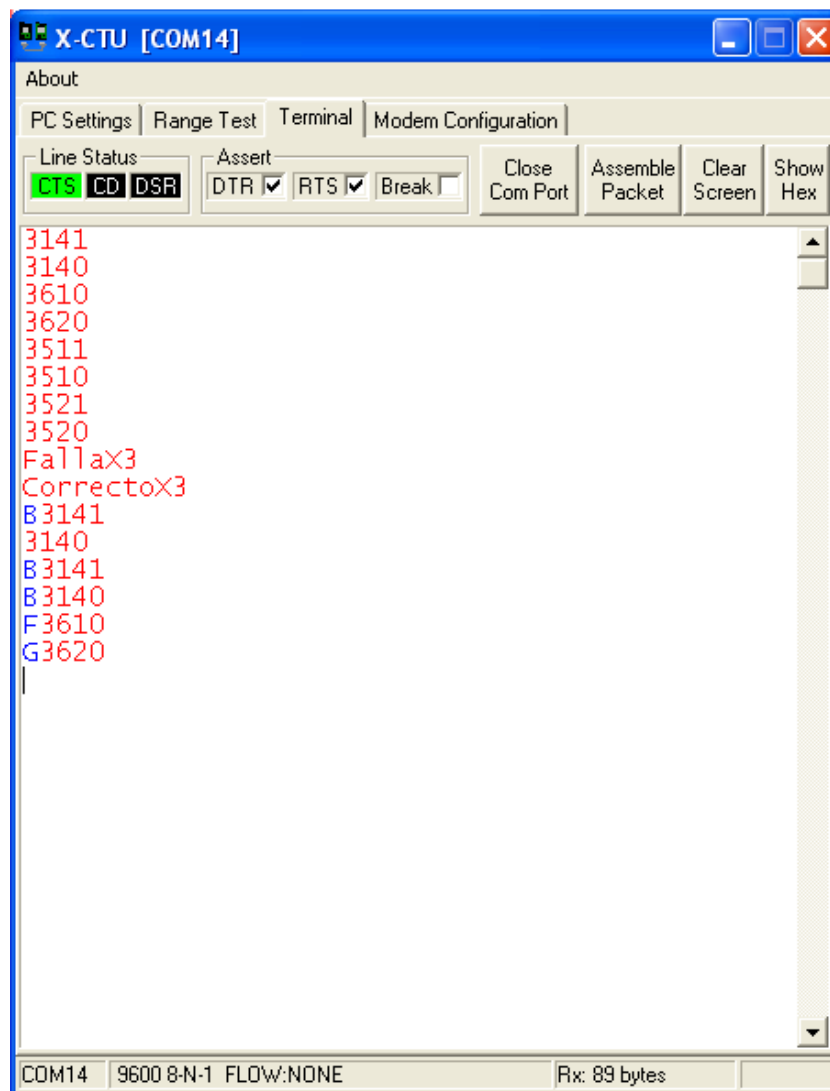


Figura 247: Pruebas funcionamiento Módulo Xbee 3

### 6.5.5. MÓDULO GLCD

Las pruebas en el módulo GLCD se realiza de manera similar al Bluetooth, Se reciben comandos enviados a un puerto COM y son visualizados por un Hyperterminal.

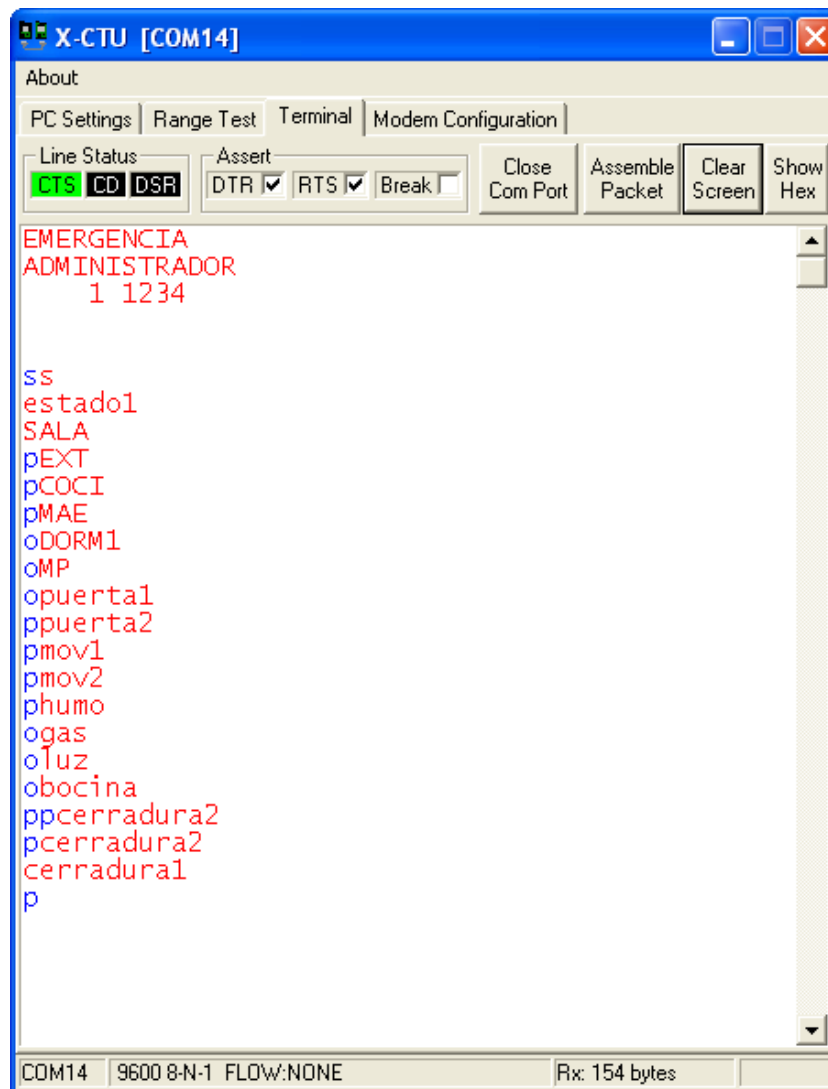


Figura 248: Pruebas funcionamiento GLCD

## CONCLUSIONES

- El desarrollo de una red mediante dispositivos que cuenten con tecnología Zigbee, es hoy en día una manera eficiente, económica y sobre todo práctica de construir un sistema domótico, que abarque tres puntos básicos seguridad, disponibilidad y convergencia.
- La creación y administración de la red Mesh/Zigbee creada, aplica todos los conocimientos adquiridos durante la carrera universitaria y no se enfoca específicamente en uno solo, esto servirá como un recuento de todo lo vivido y hará un enfoque claro de la realidad que hoy en día se vive en el mundo profesional.
- El dar soluciones efectivas a un problema en específico mediante el ingenio es la base que persigue todo profesional, es así como el desarrollo de éste proyecto presenta una ingeniosa solución a un problema en particular que es la inseguridad.
- El uso en conjunto de varias tecnologías no es complicado, sino todo lo contrario crea nuevas salidas para el desarrollo de un sistema y no lo limita, abriendo las puertas a soluciones más rápidas y eficientes.
- Las herramientas utilizadas para el desarrollo del sistema fueron de gran utilidad y trascendencia al momento de crear un sistema eficiente, estable y sobretodo seguro, los modos de seguridad implementados fueron estudiados con mucho cuidado, tratando de no dejar lugares desprotegidos, enfocándose siempre en las metas planteadas al principio del proyecto.
- El proyecto final satisface y alcanza todos los objetivos que en un principio se establecieron, y en algunos puntos superó las expectativas,



debido a que en el transcurso del desarrollo fue necesario implementar nuevos módulos para obtener un resultado más eficiente.

- Las tecnologías inalámbricas ocupan gran espacio en el diario vivir, todo las personas requieren estar siempre en continua comunicación en el lugar en que se encuentren y no verse atados a sitios donde no se puedan movilizar libremente, la solución a este problema son este tipo de tecnologías, que generan esa libertad de manera eficiente y con dispositivos portátiles de fácil manejo.

## RECOMENDACIONES

- El manejo de contraseñas de acceso al sistema debe ser manejado de manera reservada y con personas de confianza absoluta, tomando en cuenta que es un sistema de seguridad, que si no es manejado con precaución, se corre el riesgo de dejar sin protección el lugar y volviéndolo vulnerable.
- Como el sistema es dirigido a la seguridad de un hogar se recomienda que las claves de acceso tanto para los medios vía Bluetooth, SMS y Web sean manejadas por persona adultas, que tengan un criterio mucho mas formado y comprendan las consecuencias que se presentaría si el sistema es manejado de manera irresponsable.
- En los casos que los usuarios del sistema se encuentren fuera del hogar protegido se recomienda siempre dejar activado el sistema en Modo alarma Sonora, modo simulador de presencia y activar las notificaciones vía correo y SMS de los eventos que sucedan con los sensores.
- El uso de un sistema de alimentación interrumpida es esencial en el servidor del sistema es por eso que se recomienda el uso de un UPS y en el mejor de los casos uno que genere un mayor lapso de tiempo de respuesta contra fallas eléctricas, y de esta forma tener mayor tiempo para tomar nuevas medidas de seguridad.
- La cámara de vigilancia debe ser colocada en un lugar que abarque mayor grado de visibilidad del hogar y con alta iluminación, que centre su atención en lugares por los cuales exista algún tipo de vulnerabilidad, y no en lugares cerrados.

- El sistema cuenta con opciones de seguridad para el respaldo de la base de datos, es aconsejable tener éstos respaldos en diferentes medios de almacenamiento y realizarlos con frecuencia.

# **BIBLIOGRAFÍA**

## **TEXTOS**

1. CORRALES V. SANTIAGO, “Electrónica Práctica con Microcontroladores PIC”, Agosto 2006, Ecuador
2. ANGULO IGNACIO, ECHEVERRÍA MIKEL Y ANGULO JOSÉ MARÍA, “Pantallas gráficas LCD con panel táctil. Tutorial de manejo y programación”, Noviembre 2008
3. KITO D. MANN, “Java Server Faces in Action”, Manning Publications Co.,
4. TOJEIRO CALAZA GERMÁN, “PROTEUS Simulación de circuitos electrónicos y microcontroladores a través de ejemplos”, Primera Edición, Gráficas Díaz, 2009
5. REYES A. CARLOS, “Microcontroladores PIC Programación en Basic”, Segunda Edición, Quito-Ecuador 2006
6. GARCÍA BREIJO EDUARDO, “Compilador C CCS y simulador PROTEUS para microcontroladores PIC”, Segunda Edición, Editorial Marcombo, Barcelona –España 2009
7. JACOBI JONAS, FALLOWS R. JOHN, “Pro JSF and Ajax”, New York
8. JOLIF CHRISTOHE, “Advanced Graphics Components Using JavaServer Faces Technology” 2006
9. EGUÍLUZ PÉREZ JAVIER, “Introducción a JavaScript”, 2007
10. ANGEL ESTEBAN, “Programación en Java”, Editorial Paco Marín,

## **PAGINAS WEB**

11. ROSEINDIA 2008, <http://www.roseindia.net/jsf/>
12. MUNDO BYTE 2008, <http://mundobyte.wordpress.com/2008/07/28/login-con-java-server-faces/>

13. CHRISTMO'S BLOG, 2007-2010,  
<http://christmo99.wordpress.com/2008/04/09/pool-connection/>
14. <http://www.javaguicodexample.com/javawebjsfjpamysqldatabaseprint13ver2.html>
15. VBULLETIN SOLUTIONS, 2010,  
<http://www.zero13wireless.net/foro/showthread.php?1483-ZigBee>
16. EMBER CORPORATION, 2007-2008, <http://portal.ember.com/node/731>
17. PARALLAX INC., 2010,  
<http://www.parallax.com/Store/Accessories/CommunicationRF/tabid/161/ProductID/679/List/0/Default.aspx?SortField=ProductName,ProductName>
18. Robots Argentina, 2007-2009,  
[http://axxon.com.ar/rob/Sensores\\_general.htm#magnetismo](http://axxon.com.ar/rob/Sensores_general.htm#magnetismo)
19. [www.digi.com](http://www.digi.com)
20. [www.microchip.com](http://www.microchip.com)
21. D-LINK LATIN AMERICA, 2010,  
<http://www.dlinkla.com/home/productos/producto.jsp?idp=1107>
22. ZIGBEE ALLIANCE, 2010, <http://www.zigbee.org>
23. <http://www.chuidiang.com/java/herramientas/javamail/enviar-correo-javamail.php>
24. MADRID ORTIZ MARIANO,  
[http://www.gigamperios.com//index.php?option=com\\_content&task=view&id=14&Itemid=1](http://www.gigamperios.com//index.php?option=com_content&task=view&id=14&Itemid=1)
25. SILVA DIEGO, <http://wiki.netbeans.org/FragmentosConVisualJSF>
26. JAVA MEXICO COMUNIDAD DE DESARROLLADORES MEXICANOS,  
[http://www.javamexico.org/blogs/gustavo/problemas\\_para\\_encryptar\\_y\\_decryptar](http://www.javamexico.org/blogs/gustavo/problemas_para_encryptar_y_decryptar)

27. TUTORIAL: JAVA + PIC + USB (PRIMEROS PASOS),  
[http://www.ucontrol.com.ar/forosmf/tutoriales-guias-y-cursos-en-ucontrol/tutorial-java-pic-usb-\(primeros-pasos\)/?PHPSESSID=5c321f1f145b3368d0b0fd78574d0fd4](http://www.ucontrol.com.ar/forosmf/tutoriales-guias-y-cursos-en-ucontrol/tutorial-java-pic-usb-(primeros-pasos)/?PHPSESSID=5c321f1f145b3368d0b0fd78574d0fd4)
28. JPICUSB: COMUNICACION PIC+USB USANDO JAVA,  
<http://www.divideandconquer.com.ar/2009/01/jpicusb-comunicacion-pic-usb-usando-java/>
29. API JAVAX.COMM,  
[http://profesores.elo.utfsm.cl/~agv/elo329/1s05/projects/Ramirez.Diaz/utilitarios/instalacion\\_javacomm.txt](http://profesores.elo.utfsm.cl/~agv/elo329/1s05/projects/Ramirez.Diaz/utilitarios/instalacion_javacomm.txt)
30. ENVIAR COMANDOS AT POR EL PUERTO SERIE,  
<http://social.msdn.microsoft.com/Forums/es-ES/devcommes/thread/54a40755-1042-401d-a7d0-43b2b5e311ff>
31. HOW TO USE GIOVYNETSERIALPORT,  
[https://docs.google.com/Doc?docid=dcp2ngv2\\_240dccsvfd5&hl=en](https://docs.google.com/Doc?docid=dcp2ngv2_240dccsvfd5&hl=en)
32. CAPRILE SERGIO, SENIOR ENGINEER,
33. [http://www.cika.com/newsletter/archives/CTC-059\\_XBeeZBSerie.pdf](http://www.cika.com/newsletter/archives/CTC-059_XBeeZBSerie.pdf)
34. INSTALACIÓN Y CONFIGURACIÓN DE APACHE TOMCAT 6.0 EN WINDOWS XP, <http://jcalderon.wordpress.com/2008/01/04/instalacion-y-configuracion-de-apache-tomcat-60-en-windows-xp/>
35. DESARROLLO DE UNA APLICACIÓN WEB JSF VISUAL,  
<http://netbeans.org/kb/docs/web/helloweb.html>
36. REDES ZIGBEE (I). INTRODUCCIÓN,  
<http://www.blogelectronica.com/redes-zigbee-i-introduccion/>
37. ING. LAFEBRE GEOVANNI, <http://micros.mforos.com/1149902/6018988-como-envio-caracteres-ascii-desde-un-pic-a-una-pc/>
38. HOJA DE DATOS CONTROLADOR - T6963C,  
<http://www.futurlec.com/LED/LCD240X128.shtml>

- 39. MOBILE PROCESSING GROUP, <http://mobile.processing.org/>
- 40. COMO MANEJAR PUERTO PARALELO Y SERIE CON JAVA,  
<http://www.forosdeelectronica.com/f26/manejar-puerto-paralelo-serie-java-663/>
- 41. APLICACIÓN WEB CON NETBEANS Y VISUAL WEB JAVASERVER  
FACES, <http://luauf.com/2008/08/05/ejemplo-de-aplicacion-web-con-netbeans-y-visual-web-javascript-faces/>
- 42. SEGURIDAD INFORMÁTICA,  
<http://foro.elhacker.net/printpage.html;topic=161120.0>
- 43. XBEE CONFIGURATION TUTORIAL, <http://www.xbee.cl/ejemplos.html>
- 44. NETBEANS 6.5.1 Y VISUAL WEB JSF - PARTE 1,  
<http://inkahack.blogspot.com/2009/05/trabajando-con-netbeans-651-y-visual.html>
- 45. RECARGAR TXT EN JAVA WEB JSF CADA T SEGUNDOS,  
<http://www.forodejava.com/f14/recargar-txt-en-java-web-jsf-t2650/>
- 46. DESARROLLO WEB Y JAVA, <http://www.taringa.net/posts/ebooks-tutoriales/1363858/Desarrollo-Web-y-Java.html>

## GLOSARIO DE TÉRMINOS

**SMS:** Servicio de mensajes cortos Short Message Service

**PDA:** Personal Digital Assistant – Asistente Digital Personal

**Mesh:** Malla

**IEEE:** Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y Electrónicos

**PAN:** Personal Area Network - *Red* de área Personal

**OSI:** Open System Interconnection - Interconexión de Sistemas Abiertos

**Kbps:** Kilobites por segundos

**NWK:** Network Layer – Capa de Red

**APL:** Application Layer – Capa Aplicación

**ZDO:** Zigbee Device Objects – Dispositivo de Objeto Zigbee

**ED:** Energy Detection – Deteccion de Energía

**CCA:** Clear Channel Assessment – Evaluacion del canal vacío

**WiFi:** Wireless Fidelity

**Broadcast:** Difusión

**ISM:** Industrial, Scientific and Medical – Industrial, Científica y Medica

**API:** Application Programming Interface - Interfaz de programación de aplicaciones

**AT:** Attention - Atender

**VCC:** Voltaje en corriente directa

**GND:** *Ground – Tierra*

**Vin:** Voltaje de entrada



**Vout:** Voltaje de Salida

**EE:** Encryption Enable – Encriptacion Activada

**TTL:** transistor-transistor logic – Lógica Transistor Transistor

**AES:** Advanced Encryption Standard – Estándar de Encriptación Avanzado

**EO:** Encryption Options – Opciones de Encriptación

**GSM:** *Global System for Mobile communications* - Sistema Global para las comunicaciones Móviles

**E/S:** Entrada Salida

**CPU:** Unidad Central de Proceso

**CISC:** Complex Instruction Set Computer - Computadora de Conjunto de Instrucciones Complejo

**RISC:** Reduced Instruction Set Computer - Computadora con Conjunto de Instrucciones Reducido

**SISC:** Computadores con Conjunto de Instrucciones Específicas

**IP:** Protocolo de Internet

**GLCD:** Graphic Liquid Crystal Display - Pantalla Gráfica de Cristal Líquida o GLCD

**BDD:** Base de Datos

**SQL:** structured query language - Lenguaje de consulta estructurado

**PHP:** Hypertext Pre-Processor Pre Procesador de Hipertexto

**JRE:** Java Runtime Environment, - Entorno en Tiempo de Ejecución de Java

**IIS:** Internet Information Server – Servidor de Información de Internet

**AJAX:** Asynchronous JavaScript And XML – JavaScript asíncrono y XML

**IDE:** Integrated Development Environment - Entorno integrado de desarrollo

**LDR:** *light dependent resistor – Resistencia dependiente de luz*

**ADSL:** Asymmetric Digital Subscriber Line - Línea de Abonado Digital  
Asimétrica

**USB:** Universal Serial Bus – Bus Serial Universal

**JSF:** Java Server Faces

**GLP:** Gas Licuado de Petroleo

**PCB:** Printed Circuit Board – Placa de Circuito Impreso

**JAR:** Java Archive – Arcivo de Java

# **ANEXOS**

## Anexo 1

### Lugar de Implementación de Proyecto



## **Anexo 2**

### **Lista de Comandos AT**

#### **1 Comandos generales**

- a) AT+CGMI: Identificación del fabricante
- b) AT+CGSN: Obtener número de serie
- c) AT+CIMI: Obtener el IMSI.
- d) AT+CPAS: Leer estado del modem

#### **2. Comandos del servicio de red**

- a) AT+CSQ: Obtener calidad de la señal
- b) AT+COPS: Selección de un operador
- c) AT+CREG: Registrarse en una red
- d) AT+WOPN: Leer nombre del operador

#### **3. Comandos de seguridad:**

- a) AT+CPIN: Introducir el PIN
- b) AT+CPINC: Obtener el número de reintentos que quedan
- c) AT+CPWD: Cambiar password

#### **4. Comandos para la agenda de teléfonos**

- a) AT+CPBR: Leer todas las entradas

b) AT+CPBF: Encontrar una entrada

c) AT+CPBW: Almacenar una entrada

d) AT+CPBS: Buscar una entrada

## **5. Comandos para SMS**

a) AT+CPMS: Seleccionar lugar de almacenamiento de los SMS

b) AT+CMGF: Seleccionar formato de los mensajes SMS

c) AT+CMGR: Leer un mensaje SMS almacenado

d) AT+CMGL: Listar los mensajes almacenados

e) AT+CMGS: Enviar mensaje SMS

f) AT+CMGW: Almacenar mensaje en memoria

g) AT+CMSS: Enviar mensaje almacenado

h) AT+CSCA: Establecer el Centro de mensajes a usar

i) AT+WMSC: Modificar el estado de un mensaje.

### Anexo 3

#### Especificaciones técnicas Nokia 3220b



General	
lanzamiento	Verano de 2004
bandas de operación	GSM 900 / GSM 1800 / GSM 1900
dimensiones (mm)	104,5 x 44,2 x 18,7
peso (gr)	86
volumen (cc)	[ información no disponible ]
batería	Estándar BL-5B Li-Ion 760 mAh
tiempo de carga (h)	[ información no disponible ]
autonomía en espera (h)	280
autonomía en conversación (h)	3
certificación tasa específica de absorción (SAR)	Si

Memoria	
memoria en SIM	250
memoria en el teléfono	256
memoria dinámica	4 MB de memoria compartida
llamadas realizadas	10
llamadas recibidas	10
llamadas perdidas	10
memoria sms	4 MB de memoria compartida
memoria aplicaciones java	4 MB de memoria compartida
memoria imágenes	[ información no disponible ]
memoria calendario	4 MB de memoria compartida
memoria agenda/tareas	4 MB de memoria compartida
memoria comandos de voz	[ información no disponible ]
memoria marcación por voz (números)	[ información no disponible ]
memoria recordatorio vocal (segundos)	[ información no disponible ]
templates SMS	[ información no disponible ]
ampliación de memoria	[ información no disponible ]

Sistema Operativo	
sistema operativo	[ información no disponible ]
MIDP	Si
descarga de aplicaciones java	Si
otras características	<ul style="list-style-type: none"> <li>Java MIDP 2.0</li> </ul>

Conectividad	
modem integrado	Si
velocidad	Hasta 48 Kbps
infrarojos	No



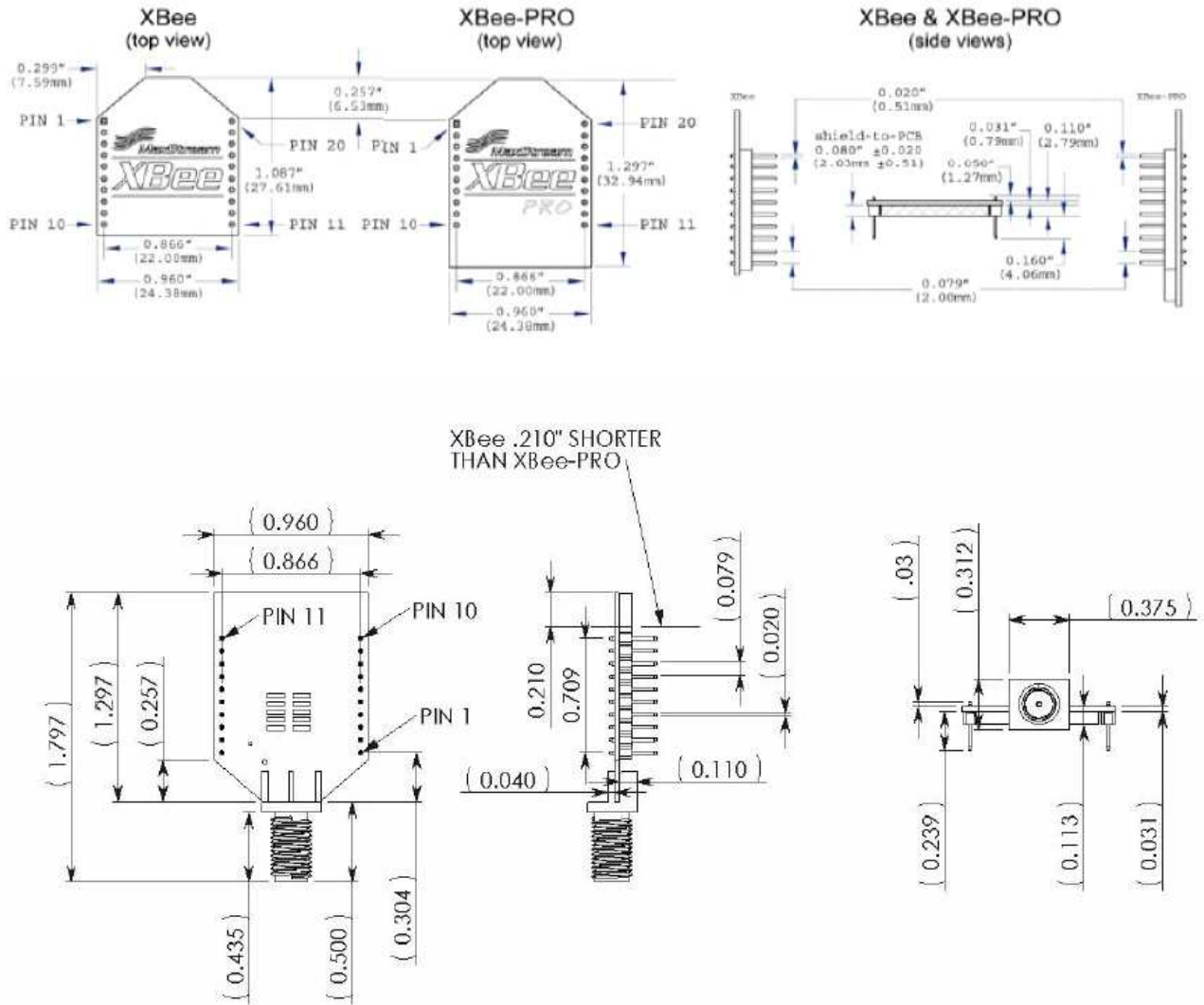
<b>bluetooth</b>	No
<b>sincronización con pc</b>	Si
<b>tipo sincronización con pc</b>	PC suite para Nokia 3220
<b>sincronización con terminales móviles</b>	No
<b>tipo sincronización con terminales móviles</b>	[ información no disponible ]
<b>sincronización remota (SyncML)</b>	Si
<b>Compatibilidad con...</b>	[ información no disponible ]
<b>puerto POP</b>	Si

<b>Mensajería</b>	
<b>mensajes cortos</b>	Enviar / Recibir / Difusión
<b>mensajes cortos concatenados</b>	Si
<b>plantillas SMS</b>	[ información no disponible ]
<b>mensajería multimedia</b>	MMS
<b>plantillas MMS</b>	[ información no disponible ]
<b>email</b>	Cliente POP3 / SMTP / IMAP4
<b>chat sms</b>	Si
<b>texto predictivo (T9)</b>	Si

## Anexo 4

### Dimensiones y especificaciones técnicas Módulos Xbee

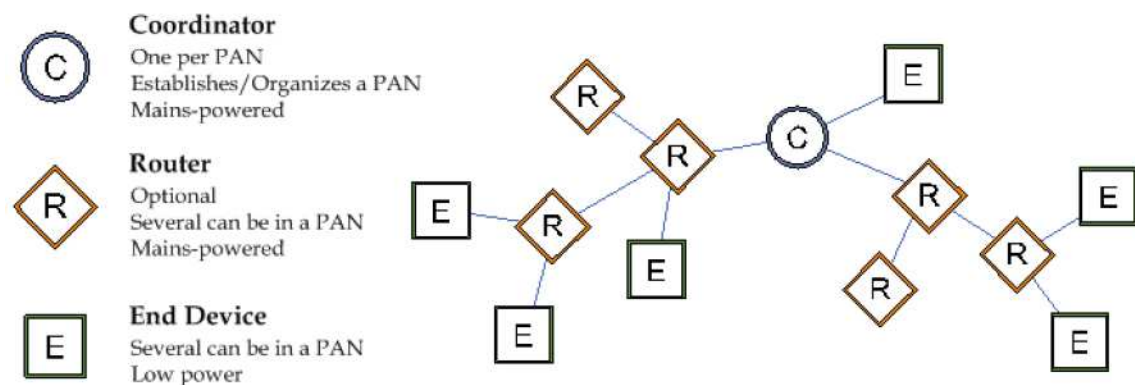
Mechanical drawings of the XBee /XBee-PRO ZNet 2.5 OEM RF Modules (antenna options not shown)



## Pin Out módulo Xbee series 2

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / <b>CONFIG</b>	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	<b>RESET</b>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	<b>DTR</b> / SLEEP_RQ/ DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	<b>CTS</b> / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / <b>SLEEP</b> / DIO9	Output	Module Status Indicator or Digital I/O 9
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	<b>RTS</b> / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

## Node Types / Sample of a Basic ZigBee Network Topology



## Datasheet GLCD

### STN Graphic LCD Display Module With Backlight

## General Specifications

Display: 240 X 128 Dots

<b>Model No.</b>	APM240128D Y/YG	APM240128D Y/JG	APM240128D B/W
<b>LCD Type</b>	STN Yellow Green	STN Yellow Green	STN Blue Negative
<b>LED Backlight</b>	Yellow Green	Green	White

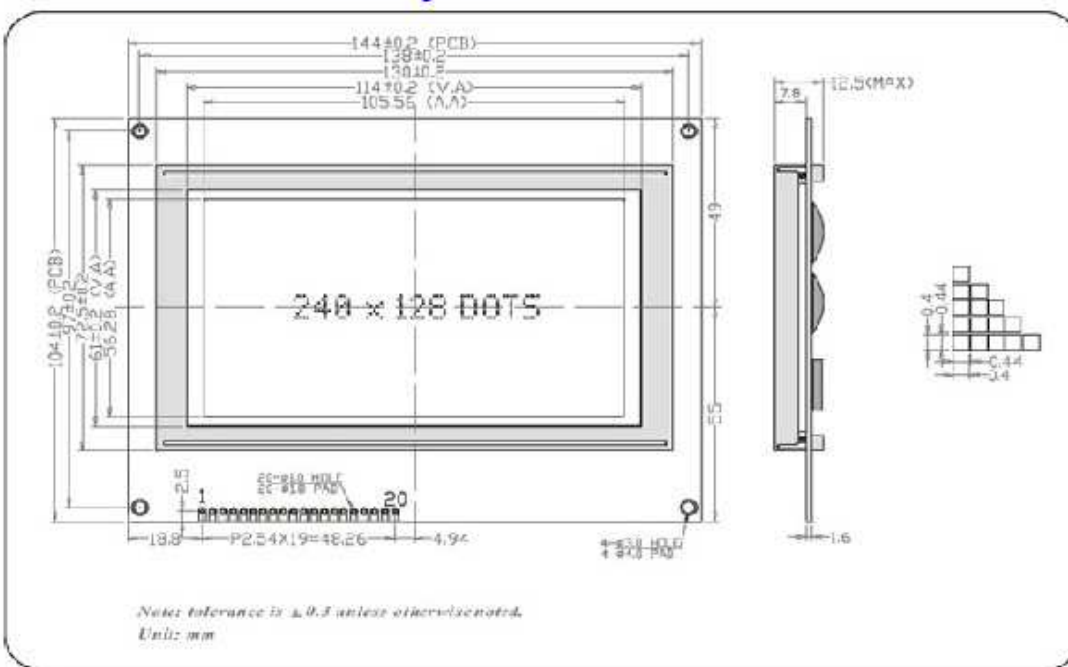
Module Size: 144.0 X 104.0 X 12.5

Viewing Area: 114.0 X 61.0

Character Size: 0.40 X 0.40

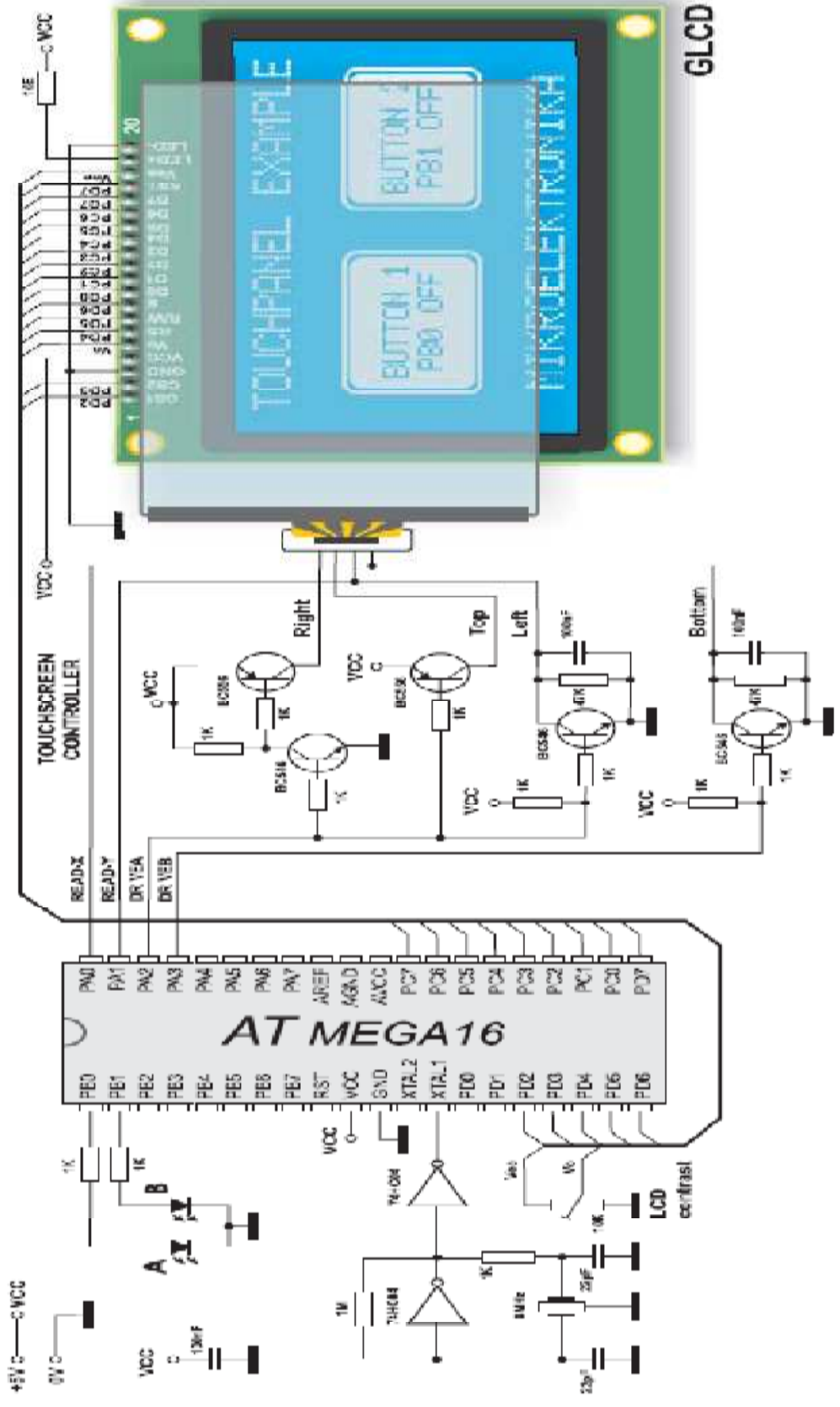
Controller IC: T6963C or equal

### Dimensions & Pin Configuration



## ■ PIN CONFIGURATION

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
FG	GND	V <sub>DD</sub>	V <sub>O</sub>	WR	RD	CE	C/D	RST	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	FS	V <sub>EE</sub>	LED



GLCD

AT MEGA16

TOUCHPANEL EXAMPLE

PIKROELEKTRONIK

## Anexo 6

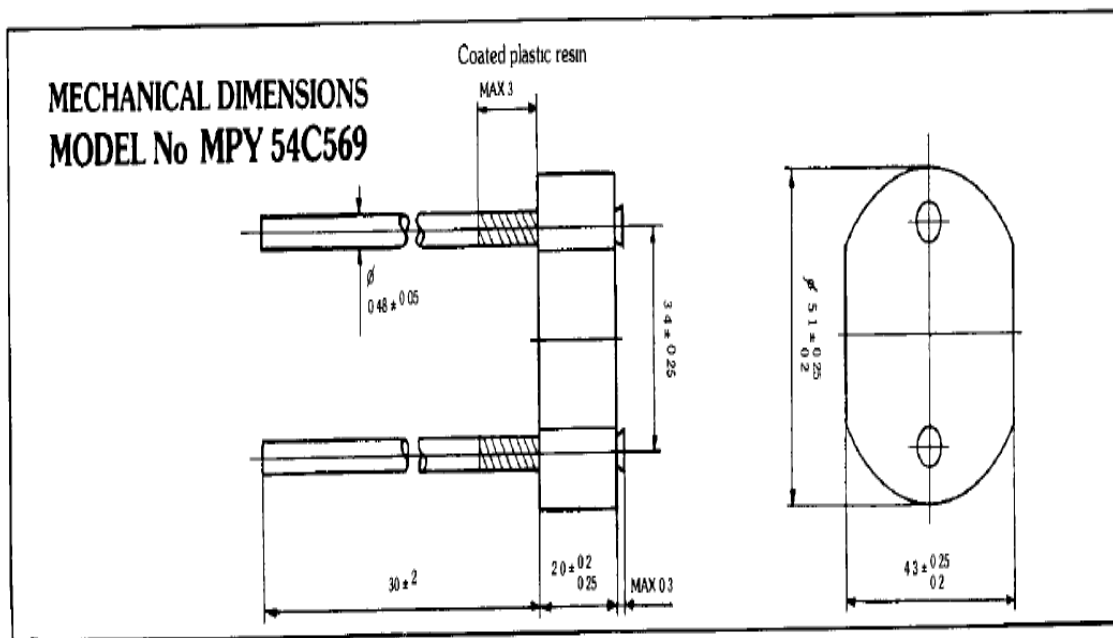
### Datasheet LDR utilizado

#### SPECIFICATIONS

Light resistance: (light source 2850°K) at 10 Lux	20 - 100 K $\Omega$
at 100 Lux	Typ. 5 K $\Omega$

\* The cell should be exposed to 400 Lux light for 2hrs prior to the above testing.

Dark resistance: (after 10 sec.)	20 M $\Omega$ Min.
Power dissipation: (Continuous) at 25°C	60 mW
(Demand) at 25°C	90 mW
Max. voltage: (DC.) in Dark	200 V
Max. spectral response:	5500 $\pm$ 300 Å
Ambient temperature range:	- 30°C ~ + 70°C
Operating temp. range:	- 30°C + 60°C



## Anexo 7

### Especificaciones técnicas del detector de movimiento

#### TS-6511 SPECIFICATIONS

Supply Voltage: 12VDC

Current Drain: 18MA Stand by

Alarm Contact: 20MA Max@ 12VDC

Tamper Switch: 1AMP Max@ 12VDC

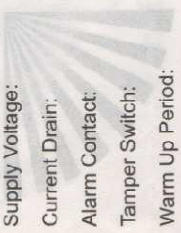
Warm Up Period: 30 Seconds

Sensitivity: Selectable Link-Normal, Medium

Mounting Height: 2.2 MTR Nominal

Vertical Beam Adjustment: Screw Adjustable

#### DETECTION PATTERN



#### SENSITIVITY

**NORMAL SENSITIVITY**  
With the jumper in this position the PIR is most sensitive. Movement into or out of a beam creating an alarm condition.

**MEDIUM SENSITIVITY**  
With the jumper in this position the PIR requires movement into and out of a beam to create alarm condition.

**LOW SENSITIVITY**  
With the jumper in this position the PIR requires movement into and out of several beams to create an alarm condition.

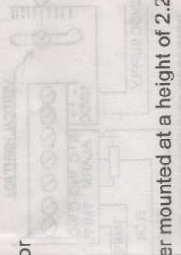
#### TS-6511 PASSIVE INFRARED DETECTOR(SMD)

The TS-6511 design incorporates highly reliable SMD circuitry excellent detection capability. Along with an exceptional high immunity to prevent form false alarms.

#### MAJOR FEATURES

- \*High PIR Rejection
- \*Selectable Sensitivity
- \*Lens Library -SMD
  - Long Range
  - Curtain

#### WIRING CONNECTIONS



#### Mounting THE DETECTOR

TS-6511 is designed to be wall or corner mounted at a height of 2.2 meters approximately. Detection zone height may be varied by vertical adjustment of the PCB draughts do not enter through any holes made in the case. Avoid pointing PIRs at heat sources, especially the sun, or white light(car headlights).



## WIRING CONNECTIONS

### + -12VDC

Connect to a regulated 12VDC out from the control unit.

### C-NC TMPR

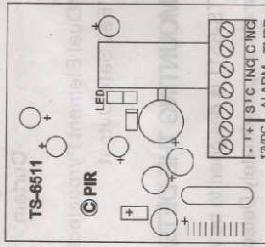
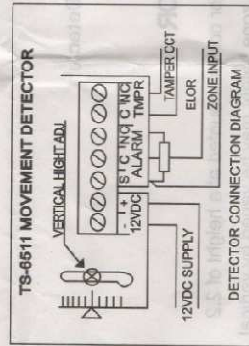
Normally closed contacts opening when the detector cover is removed.

### S TERMINAL

Spare termination which may be utilized when terminating the controls "End of line resistor"

### C-NC ALARM

Normally closed contact opening on activation or removal of 12VDC



## DETECTION PATTERNS

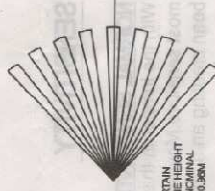
### Standard Lens

Range 15 Meters  
Viewing Angle 90°  
Detection Zones 48



### Curtain Lens

Range 12 Meters  
Beam Width 0.9 Meters  
Detection Zones 11  
(Vertical)



### Long Range Lens

Range 30 Meters  
Beam Height 3 Meters  
Detection Zones 10



LONG RANGE  
ZONES COMPARED WITH STANDARD  
2.1 METER NEUT HEIGHT

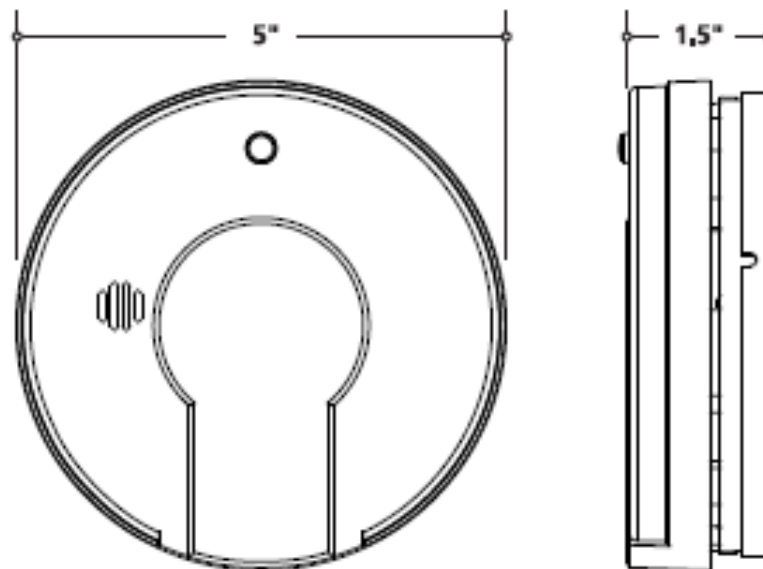


## Anexo 8

### Datasheet detector de Humo

## Technical Specifications

Model:	i9030
Power Source:	9V Battery (included)
Audio Alarm:	85dB at 10ft
Temperature Range:	40°F (4.4°C) to 100°F (37.8°C)
Humidity Range:	up to 85% relative humidity (RH)
Sensor:	Ionization
Size:	5" in diameter x 1.5" depth
Weight:	1lb
Interconnects:	No
LED:	Red, normal operation
Warranty:	3 year limited



## Anexo 9

### Especificaciones técnicas del detector de gas

#### SPECIFICATIONS

##### A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V <sub>c</sub>	Circuit voltage	5V±0.1	AC OR DC
V <sub>H</sub>	Heating voltage	5V±0.1	AC OR DC
P <sub>L</sub>	Load resistance	20K Ω	
R <sub>H</sub>	Heater resistance	33 Ω ± 5%	Room Tem
P <sub>H</sub>	Heating consumption	less than 750mw	

##### B. Environment condition

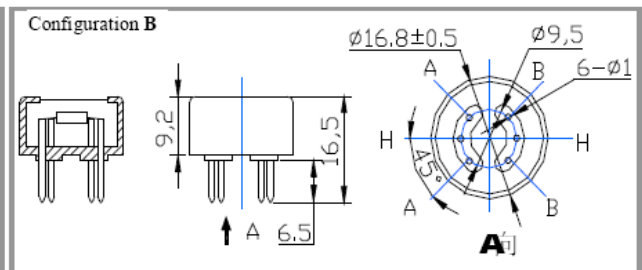
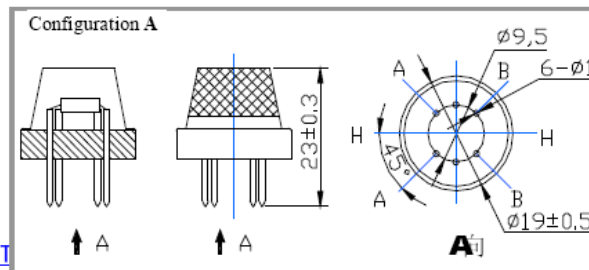
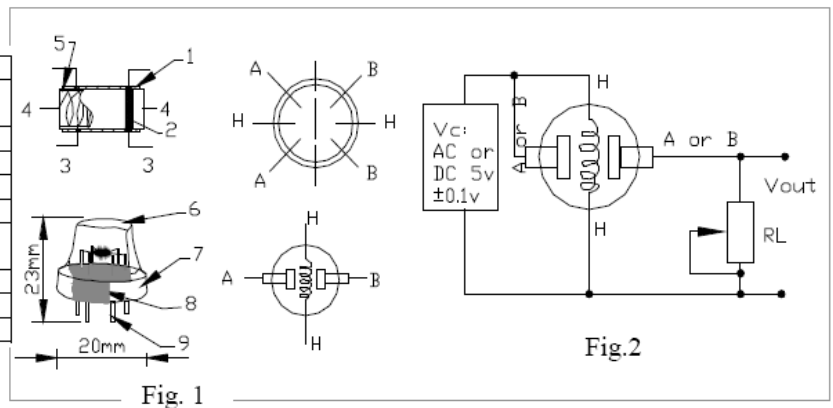
Symbol	Parameter name	Technical condition	Remarks
T <sub>ao</sub>	Using Tem	-10°C-50°C	minimum value is over 2%
T <sub>as</sub>	Storage Tem	-20°C-70°C	
R <sub>H</sub>	Related humidity	less than 95%Rh	
O <sub>2</sub>	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	

##### C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	10K Ω - 60K Ω (1000ppm LPG )	Detecting concentration scope: 200-10000ppm LPG , iso-butane, propane, LNG
α (1000ppm/ 4000ppm LPG)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃±2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

##### D. Strucyure and configuration, basic measuring circuit

Parts	Materials
1 Gas sensing layer	SnO <sub>2</sub>
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al <sub>2</sub> O <sub>3</sub>
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni



## Anexo 10

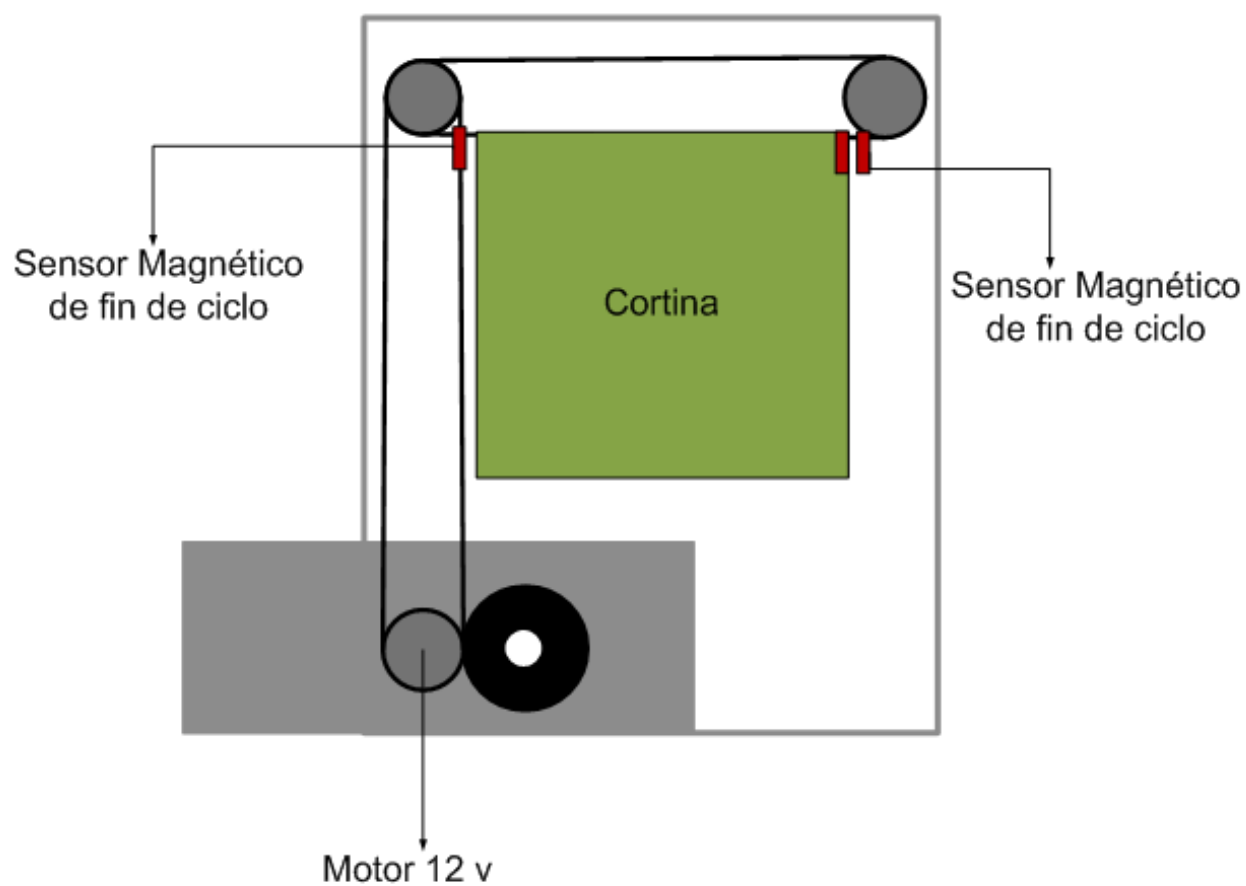
### Ficha Técnica Cámara IP

FICHA TECNICA	
INTERFACES	<ul style="list-style-type: none"><li>- 10/100BaseT ports x1</li><li>- Compliant to following standards:</li><li>- IEEE 802.3 compliance</li><li>- IEEE 802.3u compliance</li><li>- Support Full-Duplex operations</li><li>- MDI/MDIX auto-negotiation</li><li>- 802.3x Flow Control support for Full-Duplex mode</li></ul>
HARDWARE	<ul style="list-style-type: none"><li>- VGA 1/4" CMOS Sensor</li><li>- SDRAM 8 MB</li><li>- Memoria Flash 8 MB</li><li>- Soporte de Control Automático de Ganancia (AGC)</li><li>- Soporte de Balance Automático del blanco (AWB)</li><li>- Soporte Disparador Electrónico Automático (AES)</li><li>- Disparador Electrónico 1/60(1/50)~1/15,000 sec</li><li>- Mínima iluminación : 1 lux@F1.9</li><li>- Montaje del Lente: 4.57mm lens, F1.9</li></ul>
LENTE	Distancia focal 4.57mm lens, F1.9
VIDEO	<p>Modos de salida de sensor</p> <ul style="list-style-type: none"><li>- Up to 15 frames at 640x480</li><li>- Up to 30 frames at 320x240</li><li>- Up to 10 frames at 160x120</li><li>- Tamaño de imagen ajustable y calidad</li><li>- Time stamp y text overlay</li><li>- Flip &amp; mirror</li><li>- Zoom digital hasta 4X</li><li>- Soporta compresión MJPEG</li><li>- Soporta JPEG para imágenes fijas.</li></ul>

ACCESO USUARIOS	Vía Web.
PROTOCOLOS SOPORTADOS	<ul style="list-style-type: none"> <li>- IPV4, ARP, TCP, UDP, ICMP</li> <li>- DHCP Client</li> <li>- NTP Client (D-Link)</li> <li>- DNS Client</li> <li>- DDNS Client (D-Link)</li> <li>- SMTP Client</li> <li>- FTP Client</li> <li>- HTTP Server</li> <li>- PPPoE</li> <li>- LLTD</li> </ul>
SEGURIDAD	<ul style="list-style-type: none"> <li>- Default Admin ID/PWD Admin/space</li> </ul>
CARACTERISTICAS	<ul style="list-style-type: none"> <li>- Remote video monitoring and recording</li> <li>- CMOS sensor with glass lens for excellent image quality</li> <li>- 1 lux CMOS sensor can capture video in low-light environments</li> <li>- The Motion JPEG streams for remote monitoring</li> <li>- Send Snapshots to FTP or via E-mail</li> <li>- UPnP &amp; DDNS support for network configure</li> </ul>
FIRMWARE	Upgrade fail protection (Backup boot loader code)
LED'S INDICADORES	<ul style="list-style-type: none"> <li>- Enable/Disable</li> <li>- Link - Power</li> </ul>
SISTEMA OPERATIVOS SOPORTADOS	<ul style="list-style-type: none"> <li>- Windows 2000</li> <li>- Windows XP</li> <li>- Windows Vista</li> <li>- Linux</li> <li>- MAC OS X10.3 or above</li> <li>- Pocket PC</li> <li>- 3GPP Mobile Phone</li> </ul>
SOFTWARE	<ul style="list-style-type: none"> <li>- Windows 2000</li> <li>- Windows XP</li> <li>- Windows Vista</li> </ul>
ALIMENTACIÓN ELÉCTRICA	<ul style="list-style-type: none"> <li>- Input: 100-240VAC, 50/60Hz</li> <li>- Output: 5VDC, 2.5A</li> <li>- 5 VDC 2.5A. (Adaptador de alimentación eléctrica)</li> <li>- External AC-to-DC Switching Power Adapter</li> </ul>
CONSUMO	Max 4.5W
DIMENSIONES	71.9 x 110 x 37 mm sin considerar kit de montaje
TEMPERATURA OPERACIÓN	0 a 40 °C (32 a 104 °F)
TEMPERATURA ALMACENAJE	-20 a 70 °C (-4 a 158 °F)
HUMEDAD	20% a 80% RH (Humedad relativa sin condensación)
BOTÓN RESET	Reset to factory default (maintenance only)

## Anexo 11

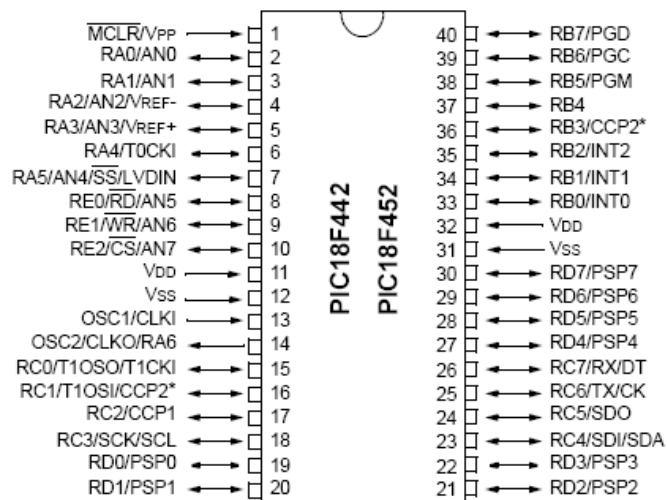
### Diagrama Prototipo Cortina Eléctrica



## Anexo 12

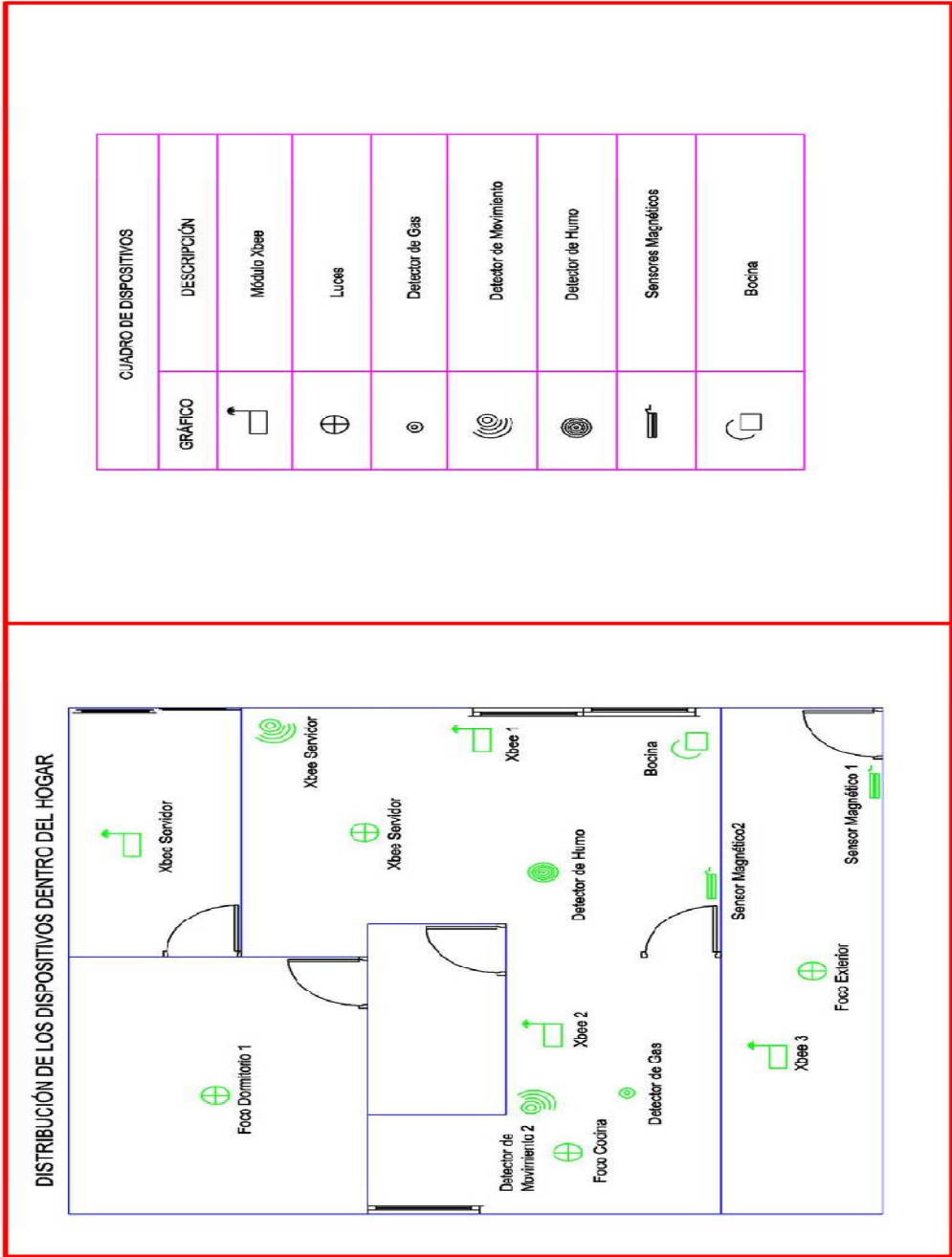
### Datasheet microcontrolador 18F452

Features	PIC18F242	PIC18F252	PIC18F442	PIC18F452
Operating Frequency	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz
Program Memory (Bytes)	16K	32K	16K	32K
Program Memory (Instructions)	8192	16384	8192	16384
Data Memory (Bytes)	768	1536	768	1536
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	17	17	18	18
I/O Ports	Ports A, B, C	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART	MSSP, Addressable USART
Parallel Communications	—	—	PSP	PSP
10-bit Analog-to-Digital Module	5 input channels	5 input channels	8 input channels	8 input channels
RESETS (and Delays)	POR, BOR, <b>RESET</b> Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, <b>RESET</b> Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, <b>RESET</b> Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, <b>RESET</b> Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions	75 Instructions	75 Instructions	75 Instructions
Packages	28-pin DIP 28-pin SOIC	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin PLCC 44-pin TQFP	40-pin DIP 44-pin PLCC 44-pin TQFP

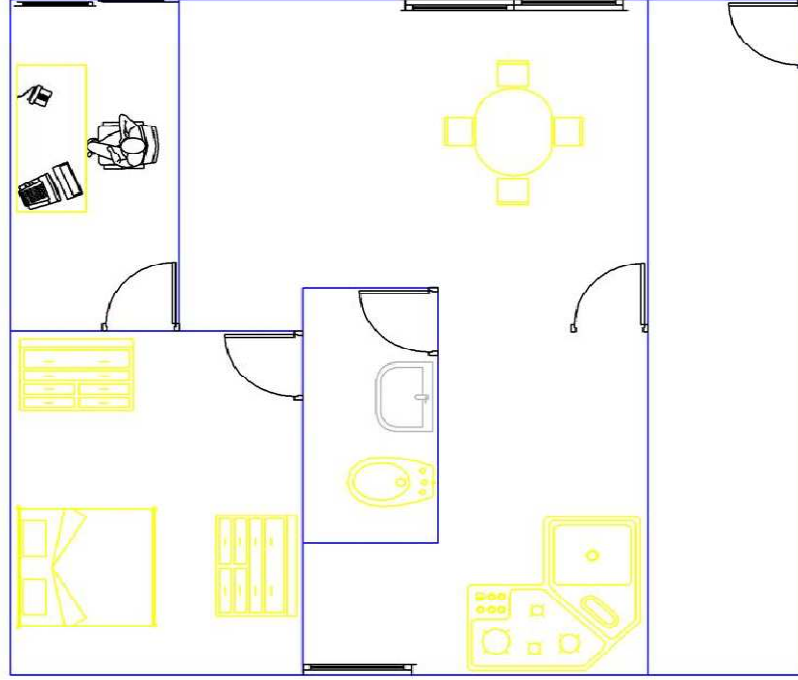


Anexo 13

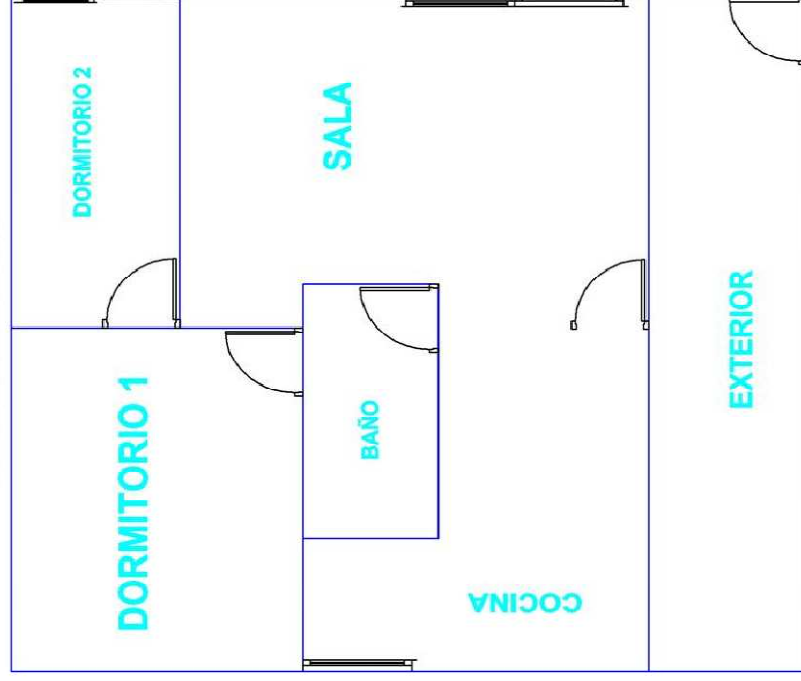
Planos de distribución de los Xbee



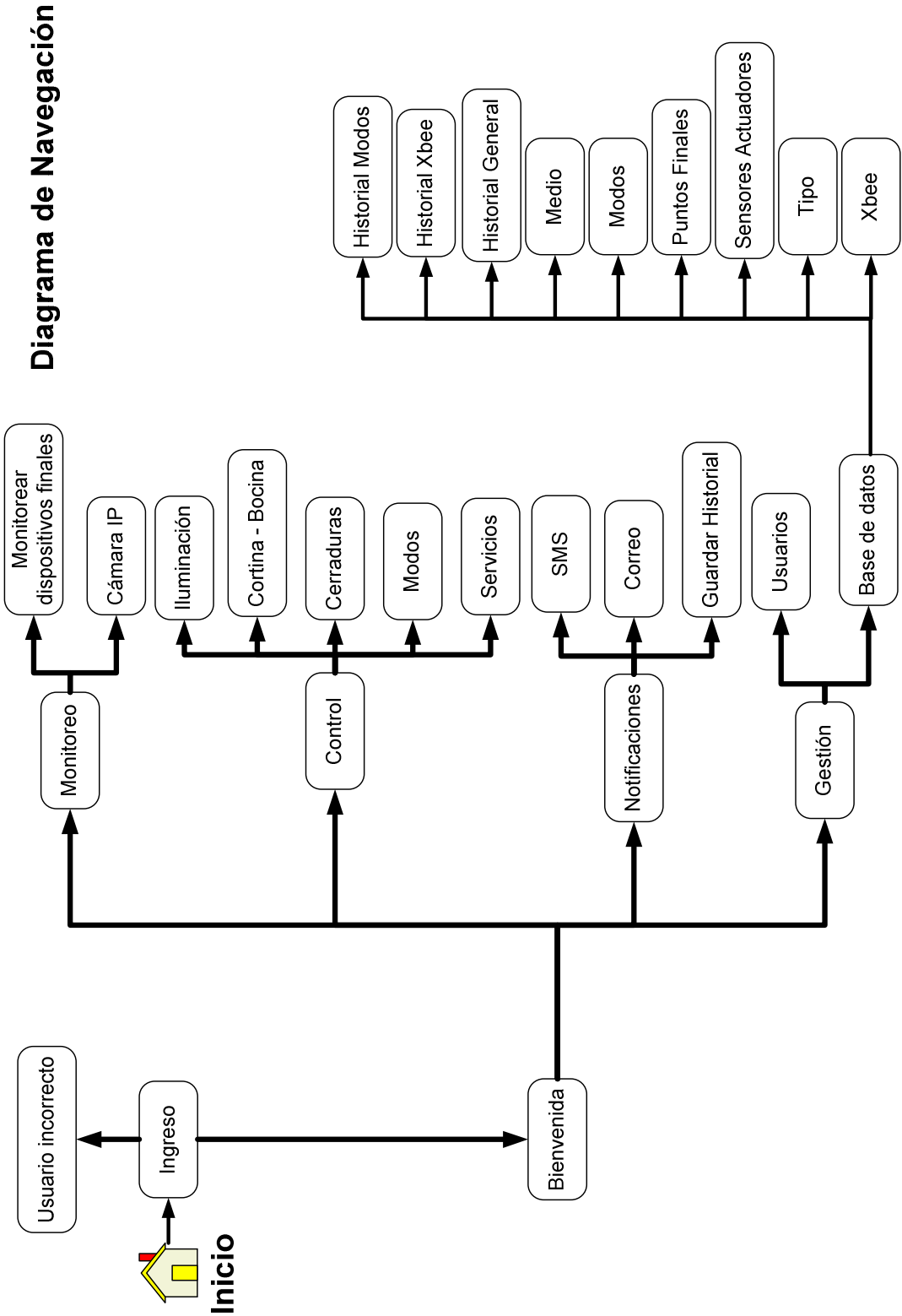
INTERIOR DEL HOGAR



DEPENDENCIAS DEL HOGAR

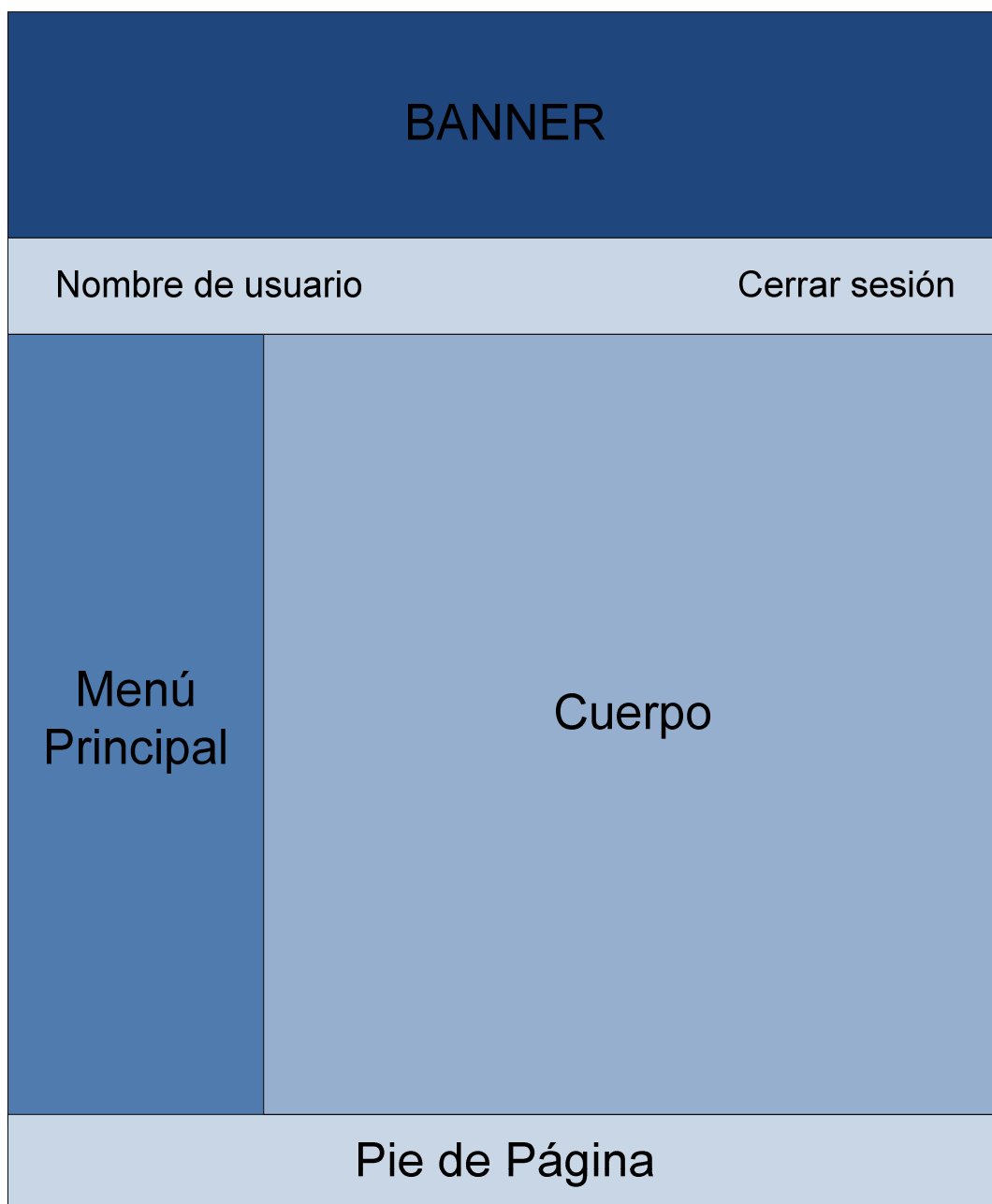






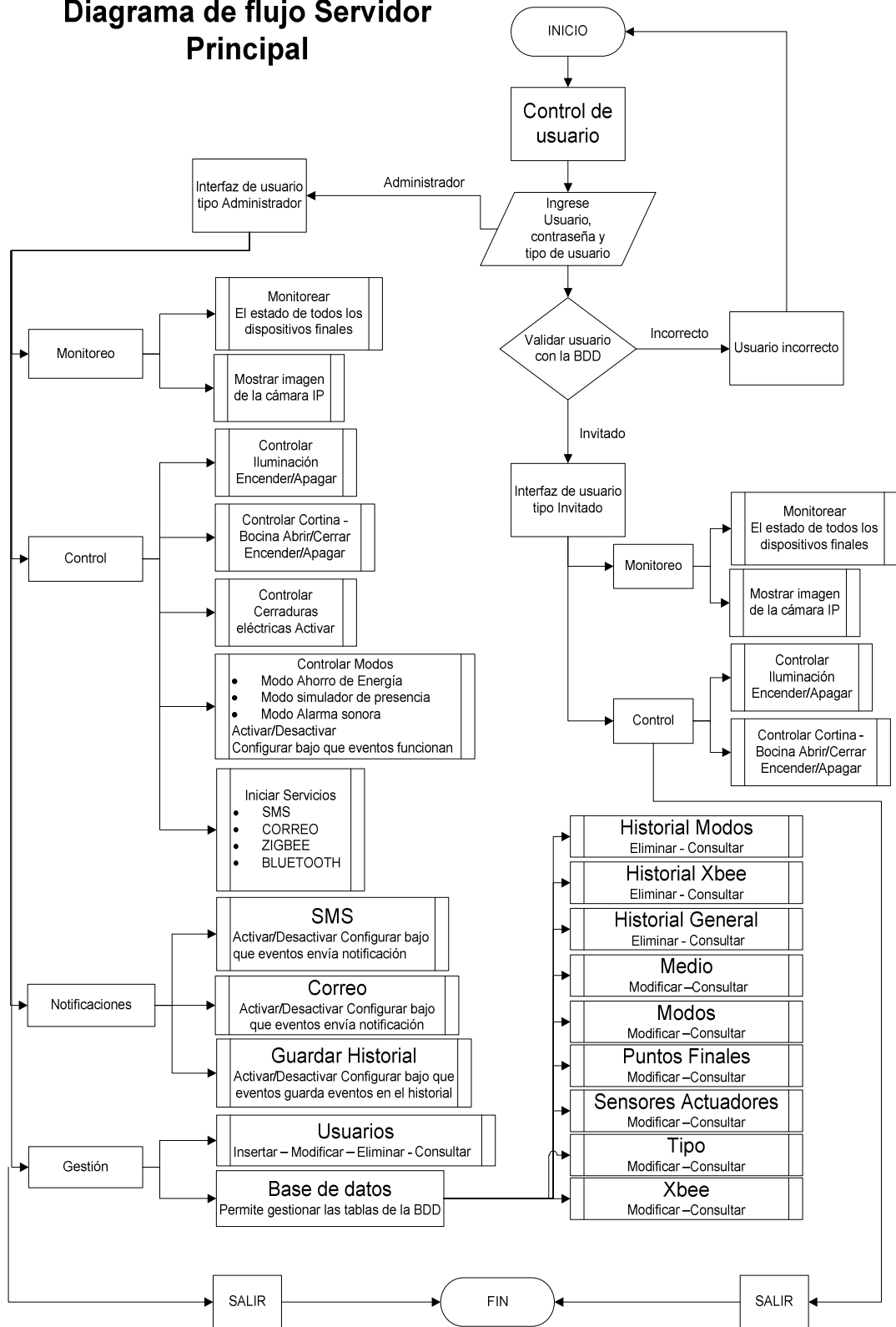
## Anexo 15

### Estructura de Páginas



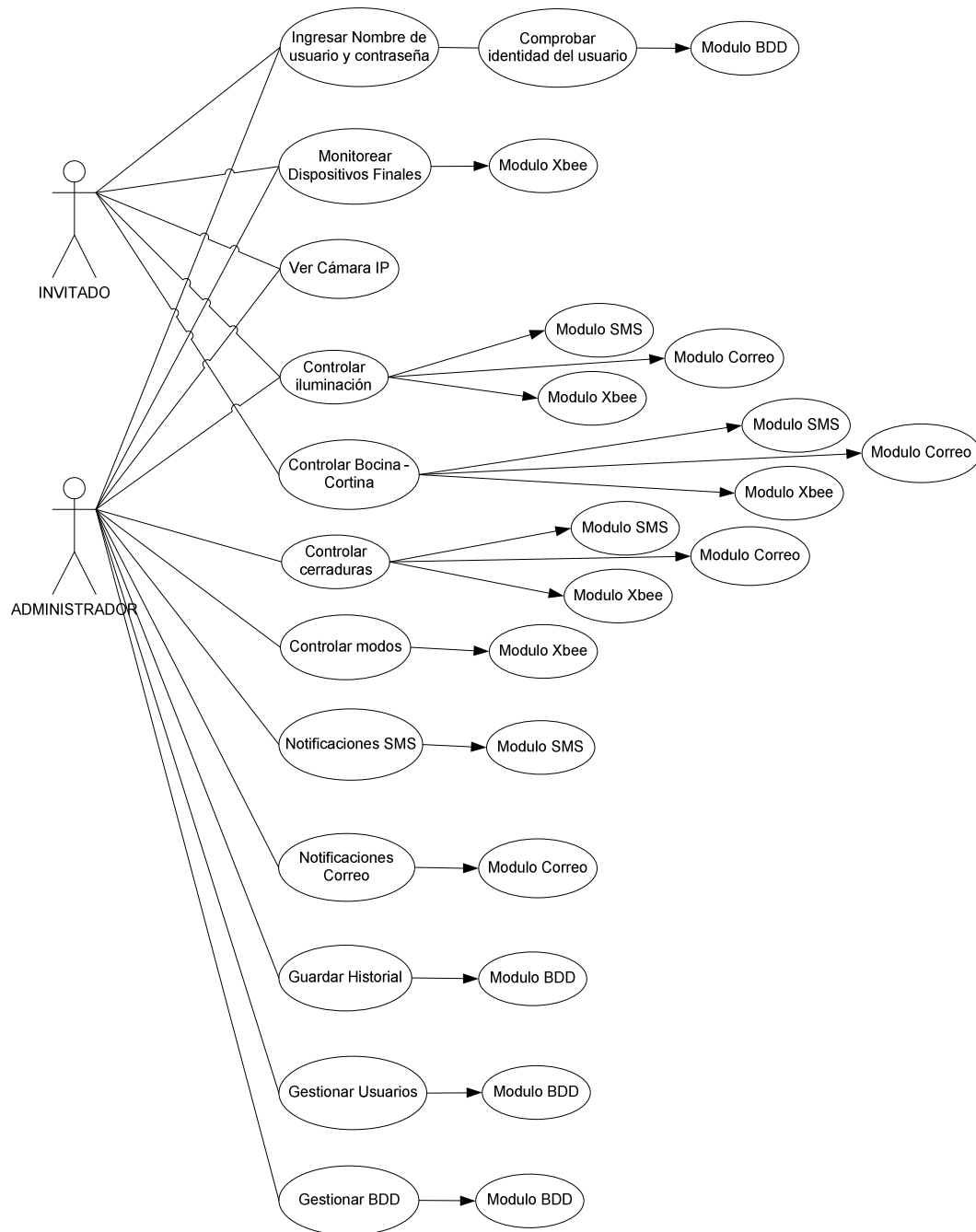
## Anexo 16

### Diagrama de flujo Servidor Principal



## Anexo 17

### CASOS DE USO “ Servidor principal ”



<b>Caso de uso 01</b>	
<b>Nombre:</b>	Ingresar Nombre de Usuario y contraseña
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Recibe nombre de usuario, contraseña y tipo de usuario
<b>Descripción:</b>	Almacena los datos recibidos en variables para poder ser enviados a comprobación.
<b>Objetivo</b>	Almacenar usuario y contraseña

<b>Caso de uso 02</b>	
<b>Nombre:</b>	Comprobar identidad del usuario
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Verifica los datos recibidos
<b>Descripción:</b>	Compara los datos recibidos por ingreso, con los que están almacenados en la base de datos
<b>Objetivo</b>	Comprobar identidad de usuario

<b>Caso de uso 03</b>	
<b>Nombre:</b>	Consultar BDD
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Realizar consultas a la base de datos
<b>Descripción:</b>	Realiza consultas requeridas por el usuario a la base de datos
<b>Objetivo</b>	Consultar base de datos

<b>Caso de uso 04</b>	
<b>Nombre:</b>	Monitoreo dispositivos finales
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Pedir estado de sensores y actuadores
<b>Descripción:</b>	Realiza peticiones del estado de los sensores y actuadores que requiera.
<b>Objetivo</b>	Recibir el estado de los sensores y actuadores, mostrarlos al usuario.

<b>Caso de uso 05</b>	
<b>Nombre:</b>	Modulo Xbee
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Envía dato, Recibe dato, lo interpreta y realiza una acción
<b>Descripción:</b>	Interpreta los datos recibidos por el servidor principal los interpreta, luego realiza acciones específicas como enviar comandos al coordinador Zigbee y recibir códigos del coordinador Zigbee.
<b>Objetivo</b>	Enviar , Recibir datos e interpretarlos para realizar una acción

<b>Caso de uso 06</b>	
<b>Nombre:</b>	Ver Cámara IP
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Mostrar imágenes en tiempo real de la cámara IP
<b>Descripción:</b>	El usuario puede monitorear su hogar por medio de la cámara IP.
<b>Objetivo</b>	Mostrar cámara IP

<b>Caso de uso 07</b>	
<b>Nombre:</b>	Control Iluminación
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Controla la iluminación
<b>Descripción:</b>	Envía los códigos de comunicación que controlan la iluminación.
<b>Objetivo</b>	Controlar la iluminación

<b>Caso de uso 08</b>	
<b>Nombre:</b>	Modulo SMS
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Recibe dato, lo interpreta y realiza una acción
<b>Descripción:</b>	Interpreta los datos recibidos por el servidor principal y luego realiza acciones específicas como enviar un SMS
<b>Objetivo</b>	Enviar SMS

<b>Caso de uso 09</b>	
<b>Nombre:</b>	Modulo Correo
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Interacción con la cuenta de correo
<b>Descripción:</b>	Envía correos electrónicos a direcciones especificadas
<b>Objetivo</b>	Enviar correo electrónico.

<b>Caso de uso 10</b>	
<b>Nombre:</b>	Controlar Bocina - Cortina
<b>Actor/es:</b>	Invitado , Administrador
<b>Función:</b>	Controla la bocina y la cortina
<b>Descripción:</b>	Controla la activación / desactivación de la bocina y la apertura / cierre de la cortina.
<b>Objetivo</b>	Controlar Bocina – Cortina.

<b>Caso de uso 11</b>	
<b>Nombre:</b>	Controlar Cerraduras
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar las cerraduras
<b>Descripción:</b>	Controla la activación de las cerraduras está dirigido solo al actor Administrador.
<b>Objetivo</b>	Controlar Cerraduras

<b>Caso de uso 12</b>	
<b>Nombre:</b>	Controlar Modos
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar los modos del sistema
<b>Descripción:</b>	Controla la activación / desactivación y bajo que eventos los modos funcionan.
<b>Objetivo</b>	Controlar lo modos.

<b>Caso de uso 13</b>	
<b>Nombre:</b>	Notificaciones SMS
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar las notificaciones por SMS
<b>Descripción:</b>	Controla la activación / desactivación y bajo que eventos se envían notificaciones vía SMS.
<b>Objetivo</b>	Enviar notificaciones vía SMS

<b>Caso de uso 14</b>	
<b>Nombre:</b>	Notificaciones Correo
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar las notificaciones por correo
<b>Descripción:</b>	Controla la activación / desactivación y bajo que eventos se envían notificaciones vía Correo electrónico.
<b>Objetivo</b>	Enviar notificaciones vía Correo

<b>Caso de uso 15</b>	
<b>Nombre:</b>	Guardar historial
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Controlar historial
<b>Descripción:</b>	Controla la activación / desactivación y bajo que eventos se insertan datos nuevos en la tabla historial de la BDD.
<b>Objetivo</b>	Guardar eventos en la Tabla Historial



<b>Caso de uso 16</b>	
<b>Nombre:</b>	Gestionar Usuarios
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Gestionar los usuarios del sistema.
<b>Descripción:</b>	Gestiona los usuarios del sistema guardados en la base de datos permite consultar, modificar, eliminar e insertar.
<b>Objetivo</b>	Modificar, eliminar, insertar y modificar los usuarios almacenados en la BDD.

<b>Caso de uso 17</b>	
<b>Nombre:</b>	Gestionar BDD
<b>Actor/es:</b>	Administrador
<b>Función:</b>	Gestionar BDD
<b>Descripción:</b>	Gestionar las distintas tablas de la base de Datos con el fin de consultar, modificar y eliminar sus registros.
<b>Objetivo</b>	Gestionar tablas de la BDD

## **Anexo 18**

### **Costos**

#### **Análisis Costo - Beneficio**

<b>Xbee 1</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
5	Relés 12v - 10 A 110V	0,55	2,75
9	Borneras 2 contactos	0,21	1,89
1	Borneras 3 contactos	0,3	0,3
1	Regulador 3.3v LD33CV	0,85	0,85
1	Regulador 5v LM7805	0,4	0,4
14	Resistencias 4.7 KΩ	0,02	0,28
4	Resistencias 10KΩ	0,02	0,08
9	Resistencias 330Ω	0,02	0,18
9	Diodos Rectificador 1N4007	0,07	0,63
5	Transistores 2N3904	0,04	0,2
2	Condensadores Cerámico 22pf/50v	0,07	0,14
1	Oscilador de Cristal 4MHz	0,49	0,49
1	Disipador Regulador 5v	0,45	0,45
1	Jack de Fuente	0,25	0,25
1	Broche Batería 9v	0,2	0,2
1	Batería 9v	1,16	1,16
2	Socket Xbee 10 pines	1,34	2,68
1	Pulsador 2 pines	0,11	0,11
14	Conectores hembra 2 pines	0,01	0,14
9	Leds	0,07	0,63
4	Pulsadores Grandes	0,45	1,8
1	Pic 18F452	8,26	8,26
1	Xbee serie 2	46,61	46,61
1	Zócalo Pic 40 pines	0,18	0,18
1	Detector de Movimiento	16,07	16,07
1	Detector de Humo	10	10
1	Bocina 12v CBZ-50	10	10
2	Sensores Magnéticos	2,28	4,56
1	Sensor Fotovoltaico	0,88	0,88
1	Caja Metálica	8,44	8,44
4	Postes 15mm	0,72	2,88
4	Tornillos Postes	0,07	0,28
5	Cables R/N	0,22	1,1
1	Funda Acido	0,36	0,36
1	Lámina	0,8	0,8
1	Impresión Laser de Circuito	0,25	0,25
1	Fuente 12 v 800mA	5	5
1	Fibra doble lado 10 x 15	1,7	1,7
1	Fibra 4 cm x 4 cm	0,35	0,35

		<b>Subtotal</b>	<b>133,33</b>
		<b>IVA 12%</b>	<b>16,00</b>
		<b>Total</b>	<b>149,33</b>
<b>Xbee 2</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
4	Relés 12v - 10 A 110V	0,55	2,2
4	Borneras 2 contactos	0,21	0,84
2	Borneras 3 contactos	0,3	0,6
1	Regulador 3.3v LD33CV	0,85	0,85
1	Regulador 5v LM7805	0,4	0,4
7	Resistencias 4.7KΩ	0,02	0,14
4	Resistencias 10KΩ	0,02	0,08
8	Resistencias 330Ω	0,02	0,16
6	Diodos Rectificador 1N4007 1A	0,07	0,42
2	Transistores 2N3904	0,04	0,08
2	Condensadores Cerámico 22pf/50v	0,07	0,14
1	Oscilador de Cristal 4MHz	0,49	0,49
1	Disipador Regulador 5v	0,45	0,45
1	Jack de Fuente	0,25	0,25
1	Broche Batería 9v	0,2	0,2
1	Batería 9v	1,16	1,16
2	Socket Xbee 10 pines	1,34	2,68
1	Pulsador 2 pines	0,11	0,11
8	Conectores hembra 2 pines	0,01	0,08
7	Leds	0,07	0,49
2	Pulsadores Grandes	0,45	0,9
1	Pic 18F452	8,26	8,26
1	Xbee serie 2	46,61	46,61
1	Zócalo Pic 40 pines	0,18	0,18
1	Detector de Movimiento	16,07	16,07
1	Sensor de Gas GLP	12,85	12,85
1	Buzzer Activo 5v	0,58	0,58
1	Caja Metálica	10	10
4	Postes 15mm	0,72	2,88
4	Tornillos Postes	0,07	0,28
3	Cables R/N	0,22	0,66
1	Funda Acido	0,36	0,36
1	Lámina	0,8	0,8
1	Impresión Laser de Circuito	0,25	0,25
1	Fuente 12 v 800mA	5	5
1	Fibra doble lado 10 x 15	1,7	1,7
1	Fibra 4cm x 4.5cm	0,35	0,35
		<b>Subtotal</b>	<b>119,55</b>

		<b>IVA 12%</b>	<b>14,35</b>
		<b>Total</b>	<b>133,90</b>

<b>Xbee 3</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
5	Relés 12v - 10 A 110V	0,55	2,75
7	Borneras 2 contactos	0,21	1,47
1	Regulador 3.3v LD33CV	0,85	0,85
1	Regulador 5v LM7805	0,4	0,4
10	Resistencias 4.7KΩ	0,02	0,2
3	Resistencias 10KΩ	0,02	0,06
8	Resistencias 330Ω	0,02	0,16
7	Diodos Rectificador 1N4007 1A	0,07	0,49
3	Transistores 2N3904	0,04	0,12
2	Condensadores Cerámico 22pf/50v	0,07	0,14
1	Oscilador de Cristal 4MHz	0,49	0,49
1	Disipador Regulador 5v	0,45	0,45
1	Jack de Fuente	0,25	0,25
1	Broche Batería 9v	0,2	0,2
1	Batería 9v	1,16	1,16
2	Socket Xbee 10 pines	1,34	2,68
1	Pulsador 2 pines	0,11	0,11
11	Conectores hembra 2 pines	0,01	0,11
8	Leds	0,07	0,56
3	Pulsadores Grandes	0,45	1,35
1	Pic 18F452	8,26	8,26
1	Xbee serie 2	46,61	46,61
1	Zócalo Pic 40 pines	0,18	0,18
2	Sensores Magnéticos	2,28	4,56
1	Caja Metálica	10	10
4	Postes 15mm	0,72	2,88
4	Tornillos Postes	0,07	0,28
4	Cables R/N	0,22	0,88
1	Funda Acido	0,36	0,36
1	Lámina	0,8	0,8
1	Impresión Laser de Circuito	0,25	0,25
1	Fuente 12 v 800mA	5	5
1	Fibra doble lado 10 x 15	1,7	1,7
		<b>Subtotal</b>	<b>95,76</b>
		<b>IVA 12%</b>	<b>11,49</b>
		<b>Total</b>	<b>107,25</b>

<b>Xbee 4</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
2	Borneras 2 pines	0,21	0,42
1	Regulador 3.3v LD33CV	0,85	0,85
1	Regulador 5v LM7805	0,4	0,4
4	Resistencias de 100 $\Omega$	0,02	0,08
1	Resistencias 4.7K $\Omega$	0,02	0,02
3	Resistencias 10K $\Omega$	0,02	0,06
3	Resistencias 330 $\Omega$	0,02	0,06
1	Diodos Rectificador 1N4007 1A	0,07	0,07
2	Condensadores Cerámico 22pf/50v	0,07	0,14
1	Oscilador de Cristal 4MHz	0,49	0,49
1	Disipador Regulador 5v	0,45	0,45
1	Jack de Fuente	0,25	0,25
1	Potenciómetro 10k $\Omega$	0,22	0,22
1	Batería Recargable 6v 1,3 A	5,9	5,9
2	Socket Xbee 10 pines	1,34	2,68
8	Conectores hembra 2 pines	0,01	0,08
3	Leds	0,07	0,21
1	Pulsadores Grandes	0,45	0,45
1	Pic 18F452	8,26	8,26
1	Xbee serie 2	46,61	46,61
1	Zócalo Pic 40 pines	0,18	0,18
1	GLCD Touchscreen	77,95	77,95
2	Regleta 40 pines hembra	0,75	1,5
1	Bus de Datos	0,5	0,5
1	Caja Metálica	8,44	8,44
4	Postes 15mm	0,72	2,88
4	Tornillos Postes	0,07	0,28
4	Tornillos Maquina	0,06	0,24
4	Rodela Plana	0,03	0,12
4	Cables R/N	0,22	0,88
1	Funda Acido	0,36	0,36
1	Lámina	0,8	0,8
1	Impresión Laser de Circuito	0,25	0,25
1	Fuente 12 v 2 A	5	5
1	Fibra doble lado 10 x 15	1,7	1,7
		<b>Subtotal</b>	<b>168,78</b>
		<b>IVA 12%</b>	<b>20,25</b>
		<b>Total</b>	<b>189,03</b>

<b>Xbee 5</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Valor Unitario</b>	<b>Valor Total</b>
<b>1</b>	Xbee serie 2	46,61	46,61
<b>1</b>	Xbee Explorer USB	33,62	33,62
<b>1</b>	Cable Mini USB	3,13	3,13
		<b>Subtotal</b>	<b>83,36</b>
		<b>IVA 12%</b>	<b>10,00</b>
		<b>Total</b>	<b>93,36</b>

<b>Costos Totales</b>	
<b>Interfaz USB</b>	93,36
<b>Xbee 1</b>	149,33
<b>Xbee 2</b>	133,9
<b>Xbee 3</b>	107,25
<b>GLCD</b>	189,03
<b>TOTAL</b>	<b>672,87</b>

Para obtener el costo total del proyecto a este valor hay que agregarle el precio de la cámara IP, Dispositivo Bluetooth USB y modem GSM.

<b>Costo total del Proyecto</b>	
Circuitos	672,87
Camara IP	110
Dispositivo Bluetooth USB	30
Modem GSM	30
<b>TOTAL</b>	<b>842,87</b>

A continuación se muestra el análisis costo beneficio del proyecto.

<b>Costo</b>		<b>Beneficio</b>	
<b>Costo del Proyecto</b>	842,87	<b>Pérdida por robo</b>	Incuantificable
<b>Costo mensual recurrente, conexión internet CNT 1024/512 kbps</b>	28	<b>Pérdida Incendio</b>	Incuantificable
<b>Mano de obra</b>	1200	<b>Ahorro de energía eléctrica</b>	Incuantificable
<b>Costos de instalación</b>	400		
<b>Dominio</b>	0		
<b>TOTAL</b>	<b>2470,87</b>	<b>TOTAL</b>	<b>Incuantificable</b>

Los beneficios son Incuantificables debido a que su cálculo representa muchas variables y escenarios, en un escenario de robo pérdidas son variables podrían ir desde unos pocos dólares hasta la pérdida de todos los bienes, en un escenario de incendio las pérdidas pueden ser menores o podrían abarcar hasta una pérdida total lo cual es incuantificable.

**2470,87 < Incuantificable**

**Costo < Beneficio**

El costo es mucho menor al beneficio que presenta la adquisición del sistema domótico por lo tanto es conveniente.



## Pruebas de base de datos

#	Consulta	Condición	Tablas	Fin	Resultado	Observaciones
1	INSERT INTO usuarios (COD_USUARIO,CONTRA, COD_TIPO,NOMBRE,CEL ULAR,MAIL,DESCRIPCION ) VALUES (('1344','3322','1','usuario','08 7821533','nuevo@hotmail.c om','nuevo usuarios'))		usuarios	Insertar nuevo usuario	OK	
2	SELECT * FROM tipo t, usuarios u WHERE t.cod_tipo=u.cod_tipo and u.cod_usuario='null'	t.cod_tipo=u.cod_tip o and u.cod_usuario='null'	tipo, usuarios	Consultar el tipo de usuario	OK	
3	SELECT * FROM tipo t, usuarios u WHERE t.cod_tipo=u.cod_tipo and u.cod_usuario='5678'	t.cod_tipo=u.cod_tip o and u.cod_usuario='567 8'	tipo, usuarios	Consultar el tipo de usuario	OK	
4	SELECT v FROM Vistahistorialmodos v WHERE v.modos='MSP'	v.modos='MSP'	Vistahistorial modos	Mostrar el contenido vistahistorial modos donde el modo se MSP	OK	
5	SELECT v FROM Vistahistorialmodos v WHERE v.usuario='carlos'	v.usuario='carlos'	Vistahistorial modos	Mostrar el contenido vistahistorial modos donde	OK	

				el usuario sea carlos		
6	SELECT v FROM Vistahistorial v WHERE v.fecha BETWEEN '2010-09-15' AND '2010-10-30' AND v.xbee='xbee1' AND v.estado='PRENDIDO' AND v.usuario='carlos' AND v.medio='Web' AND 1=1	v.fecha BETWEEN '2010-09-15' AND '2010-10-30' AND v.xbee='xbee1' AND v.estado='PRENDIDO' AND v.usuario='carlos' AND v.medio='Web' AND 1=1	Vistahistorial modos	Mostrar el historial según fecha, Xbee, estado, usuario y medio	OK	Consulta evaluando todos los parámetros de búsqueda
7	SELECT h FROM HistoXbee h where h.codXbee='1'	h.codXbee='1'	HistoXbee	Mostrar el historialxbee según el código Xbee	OK	
8	SELECT v FROM Vistahistorialmodos v WHERE v.fecha BETWEEN '2010-11-08' and '2010-11-10'	v.fecha BETWEEN '2010-11-08' and '2010-11-10'	Vistahistorial modos	Mostrar vistahistorial modos mediante dos fechas	OK	
9	DELETE FROM historial WHERE COD_HISTORIAL = '120'	COD_HISTORIAL = '120'		Eliminar el historial mediante el código del historial		
10	DELETE FROM histo_xbee WHERE cod_histo_xbee = '8'	cod_histo_xbee = '8'	histo_xbee	Eliminar el historial_xbee mediante el código_xbee	OK	

11	DELETE FROM histo_modos WHERE cod_histo_modo = '3'	cod_histo_modo ='3'	histo_modos	Eliminar el historial_mod os mediante el código del modo	OK	
12	UPDATE usuarios SET cod_usuario="" 1141 contra="" 8585 cod_tipo='1', nombre='nuevo', celular='085412547', mail=nuevo@hotmail.com, descripcion='nuevo usuario' WHERE cod_usuario='1254'	cod_usuario="" + lbl_ codigoauxiliar.getTe xt() + ""	usuarios	Actualizar la tabla usuarios	OK	
13	DELETE FROM usuarios WHERE nombre='pepe'	nombre='pepe'	usuarios	Eliminar el usuario mediante el nombre	OK	
14	CREATE VIEW VISTAHISTORIAL AS SELECT h.cod_historial as Historial, u.nombre as Usuario, p.ubicacion as Ubicacion, m.nombre as Medio, x.nombre as Xbee, sa.nombre as SensorActuador, h.estado as Estado, h.fecha as Fecha, h.hora as Hora FROM historial h, usuarios u,	h.cod_usuario=u.co d_usuario AND h.cod_punto=p.cod_ punto AND sa.cod_senact=p.co d_senact AND p.cod_xbee=x.cod_ xbee AND m.cod_medio=h.cod_ _medio	historial h, usuarios u, puntos_finale s p, medio m, xbee x, sensores_act uadores sa	Crear vista de la tabla historial	OK	Mostrar consulta multitabla mas comprensible

	<p>puntos_finales p, medio m, xbee x, sensores_actuadores sa WHERE h.cod_usuario=u.cod_usuario AND h.cod_punto=p.cod_punto AND sa.cod_senact=p.cod_senact AND p.cod_xbee=x.cod_xbee AND m.cod_medio=h.cod_medio</p>					
15	<p>CREATE VIEW VISTAHISTORIALMODOS AS SELECT m.cod_mod as Codigo, u.nombre as Usuario, m.nombre as Modo, hm.estado as Estado, hm.fecha as Fecha, hm.hora as hora FROM historial_modos hm,usuarios u, modos m WHERE hm.cod_usuario=u.cod_usuario AND hm.cod_mod=m.cod_mod o</p>	<p>hm.cod_usuario=u.cod_usuario AND hm.cod_mod=m.cod_mod</p>	<p>historial_modos hm,usuarios u, modos m</p>	<p>Crear vista de la tabla historial_modos</p>	OK	<p>Mostrar consulta multitabla mas comprensible</p>

## **Anexo 20**